

# Java Tutoring System with Facial and Text Emotion Recognition

Ramón Zatarain-Cabada, María Lucia Barrón-Estrada, Jorge García-Lizárraga,  
Gilberto Muñoz-Sandoval, José Mario Ríos-Félix

Instituto Tecnológico de Culiacán, Culiacán Sinaloa,  
Mexico

{rzatarain, lbarron, m86170331, m14170101, mario\_rios}@itculiacan.edu.mx

**Abstract.** This paper presents the design and implementation of an intelligent tutoring system (ITS) for teaching JAVA, which can recognize the user's emotional state through facial expressions and textual dialogues. For facial emotion recognition we implemented a neural network with WEKA library and a facial feature extractor with OPENCV library. The ITS applies a semantic algorithm (ASEM) to extract textual emotions through dialogues, which has shown a degree of assertiveness of 80% in tests for graduate students. In addition, the tutor uses a set of fuzzy rules to determine the complexity of the next exercise, considering the program implementation time, program executions and compilations, and current difficulty level.

**Keywords:** Affective computing, intelligent tutoring system, neural networks, fuzzy systems.

## 1 Introduction

In recent decades, technology has had a huge development by allowing the design of more complex programs, by using new software development techniques, new data mining algorithms, and more efficient software architectures. Likewise, the integration of the emotional state of the user with technologies like neural networks, genetic algorithms, and fuzzy logic produces in the case of educational software a new way of permeating awareness among students and software.

At the beginning Intelligent Tutoring Systems used to be implemented with traditional principles of behaviorism, moving later to other more interactive and dynamic learning theories, in which the subjects interact in virtual learning environments. Affective Tutoring Systems (ATS) are intelligent systems that incorporate the ability to recognize the emotional state of the students allowing the user to interact with exercises that stimulate their emotional state [1].

Given the growing demand for learning programming languages, it is necessary to use new tools using modern techniques such as emotion recognition, which motivate and facilitate student learning. At present, supporting tools for teaching programming languages only take into account cognitive aspects of the student, ignoring other behavior features like user's emotions. Studies have shown that emotions are closely related to cognitive processes such as learning [2].

This work integrates diverse technologies into the affective tutor named “Java Zenzei”, such as the use of a neural network (using WEKA) and a feature extractor for recognizing facial emotions, as well as a semantic algorithm (ASEM) to extract emotions from textual dialogue.

This paper is divided as follows. The second section describes related work. The third section mentions the effort done in the development of a bimodal recognizer that was used in our research. The fourth section describes the tutoring system operation. The fifth section shows results of the proposed work. Finally, the sixth section presents conclusion of the work.

## **2 Related Work**

There exists a lot of research about multimodal recognition to detect emotions taking into account different aspects. For example, Soleymani [3] combines EEG signals with eye tracking that can be considered bimodal recognition. There are also a lot of work with multimodal recognition like MERT [4] that works with face, voice, and body motion.

In another paper, Sebe, Cohen, and Huang [5] use web camera and microphone to improve the human-computer interaction, where they state that facial gestures and voice are crucial for that interaction resulting in different benefits for their research. D'Mello [6] has also contributed in the field of multimodal recognition with his project AutoTutor [7] which uses voice, body motion, and facial gestures to identify emotions.

A recent version of AutoTutor named Affective AutoTutor [10] adds capabilities to recognize the affective state of a student by using facial expressions, body language (posture) and textual dialogue recognition. It detects when the student is bored, confused or frustrated and intervenes through a pedagogical agent.

Binaly [11] describes four different text based emotion recognition techniques, Keyword spotting method, Lexical Affinity Method, Learning based method and hybrid methods.

The main contribution of this work is the use of a bimodal emotion recognition (facial and Spanish textual recognition) inside of an intelligent tutoring system for learning Java.

## **3 Bimodal Recognition of Emotion**

The emotion recognition is a complex task and to date the rates of most recognizers do not get 100% of success. So using bimodal recognition is a way to increase the rate of success in the recognition. Next we explain the structure of the Java Zenzei Tutor.

### **3.1 Java Zenzei Structure**

The system has a Log in component which identifies and authenticates the user. There's a component to get the user's current difficulty level by searching at the

database, in case the user is new, a diagnostic test module is executed, which shows a questionnaire to be solved, to determine the starting level of the user.

The system shows an exercise based on the user's current level. When the exercise is solved, the exercise's variables (exercise validation, amount of compilations and time required) are obtained, also the emotional state variables (facial and text emotion) are extracted by taking a photo of the user and asking a question about the exercise.

Next the system uses a fuzzy logic engine to obtain a new user's level by combining the current level, exercise's variables, and the emotional state variables. The process to obtain a new level is performed every time the user completes an exercise.

The ATS structure (figure 1) presents a clear idea of how the system works.

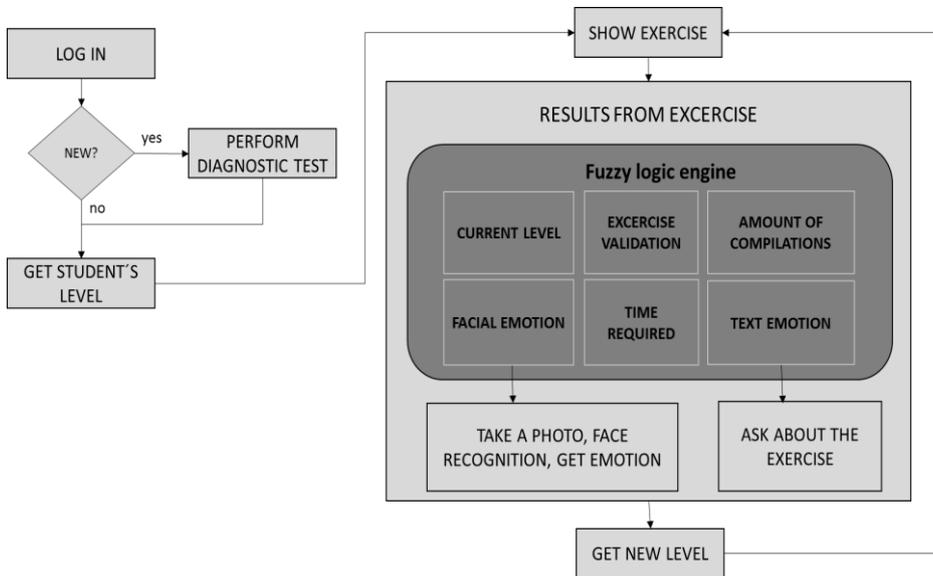


Fig. 1. ATS Structure.

Persistence of the information required by the tutor like student's information, exercises, answers for each question, information of each exercise to be executed, is stored in an object-oriented database (DB4O).

The recognition method for emotions works with two recognizers, where the first one is a method for emotion recognizing through face detection. In this process, the system take a picture of the student when he/she is solving the exercise. We use a feature extractor for the face which was implemented with OPENCV library and a neural network implemented in WEKA for classifying emotions. The emotions identified by the facial recognizer are: joy, surprise, sadness, anger and neutral emotion.

The system uses a set of fuzzy rules to identify the complexity of the next exercise where six aspects are considered: student's level (beginner, basic, intermediate, and advanced), validation of last exercise, number of compilations, identified facial emotion, time spent in exercise and identified text emotion. We used a Java library

named JFuzzyLogic for implementing the fuzzy system and a new semantic algorithm (ASEM) for textual dialogue. The evaluation allows the ATS to identify the complexity level of the next exercise which can be lesser, greater or equal to the current level. An example of a fuzzy logic rule is shown next (figure 2).

```
IF currentLevel IS intermediate AND  
   elapsedTime IS few AND  
   compilations IS regular AND  
   validation IS correct AND  
   facialEmocion IS joy AND  
   textualEmotion IS joy THEN level IS advanced;
```

Fig. 2. Fuzzy logic rule example.

The proposed emotion recognition algorithm through the text is described below.

### 3.2 ASEM Algorithm

For emotion recognition from text, we implemented a semantic algorithm (ASEM) that allows identifying emotions from text dialogues using semantic labels [8], where the user writes different comments for answering questions received in a random form. This semantic algorithm is based mainly in a word corpus called SEL [9] which was built by experts with an output emotion and a probability factor of affection (PFA).

The ASEM algorithm incorporates semantics rules that allow emotion detection with a rate of success greater than 80% according to different test performed with graduate students. The rate of success increases by adding new words together with the corresponding PFA which must be computed by an expert. The emotion recognition is made in two phases: training and application.

During the training phase, we used an interaction which allows the communication with student, where some processing text is recorded. Next the emotion is generated according to the existing words found in the original corpus, showing the parameters used in the emotion generation, and also those words which could not be found in the corpus (they are also added to the corpus *NewWords*). For each new added word an expert define the PFA. This first phase is realized in an iterative form until the new corpus has found the expected success level.

In the second phase (Application) ASEM perform the next steps (figure 3):

- A student input a text line (input dialogue).
- The text is normalized: the accented words, numbers, and special characters are removed and uppercase letters are converted to lowercase.
- Non emotion words like *he, she, the*, etc. are removed by using the corpus *StopWords*.
- Semantic words are sought in corpuses *Semantic* and *improper words*. If semantic words are not found in the corpuses the new words are added to corpus *NewWords*.

- If the semantic word is found in the corpuses, the word features (PFA and emotion) are extracted.
- The emotion is classified according to the features of the word.
- The emotion with greatest intensity is produced.

Figure 3 shows the ASEM algorithm that can recognize the emotion from text.

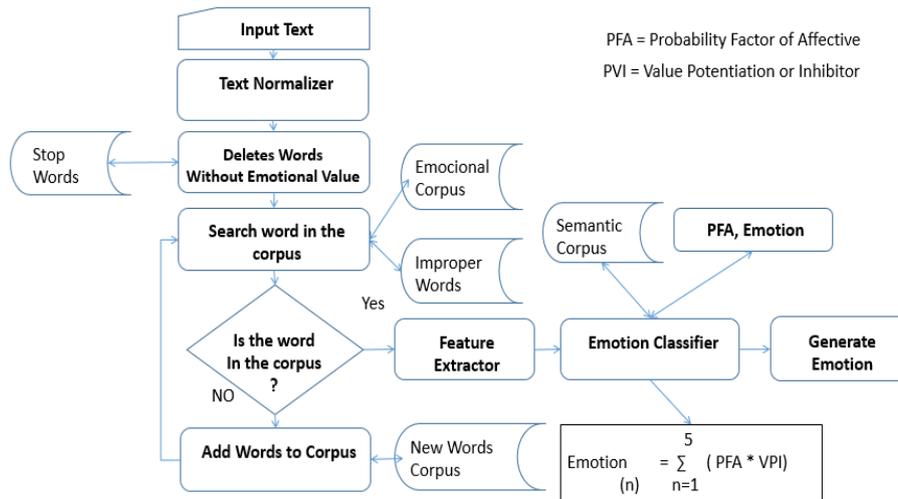


Fig. 3. ASEM algorithm.

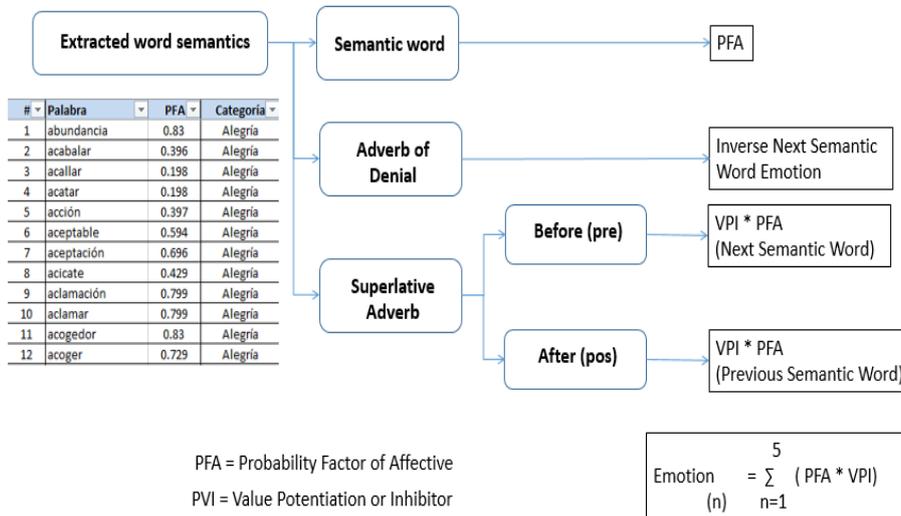


Fig. 4. Emotion detection.

To get the emotion we use PFA and EIV (enhancer/inhibitor) values for every word found in the text written by the students. The whole process is explained next (figure 4):

- The next semantic word is obtained.
- This word is sought in corpus Semantic. If found, the emotion and PFA value are extracted.
- If the word is a negative adverb, the emotion of the next word is switched (e.g. happy to sad).
- If the word is a superlative adverb, we identify the word as a pre-adverb or a post-adverb. In case of being a pre-adverb we enhance the PFA, and EIV of the next semantic word. In case of being a post-adverb we enhance the PFA and EIV of the previous last semantic word.
- Finally we use the sum of each found emotion in the text dialogue, and then we determine the emotion by the greatest value.

Next, we give some details produced in the emotion recognizing from text:

- There are many words that were not found in the original corpus (2,036 words) from a total of more than 300,000 of the Spanish language.
- Most students don't use accent in words.
- There are words with no emotion (neutral emotion). These words integrate a corpus named StopWords (e.g. *the, he, or she*)
- Improper words are eliminated from the text but they are considered as having a negative emotion.
- The emotions identified by the text recognizer are *joy, surprise, neutral, sadness and anger*.
- Words that deny a sentence (e.g. *no* or *never*) change its meaning (valence) and they are already stored in the semantic corpus.
- Words which define the intensity of the words (e.g. *very, few, nothing*) are already stored in the semantic corpus.

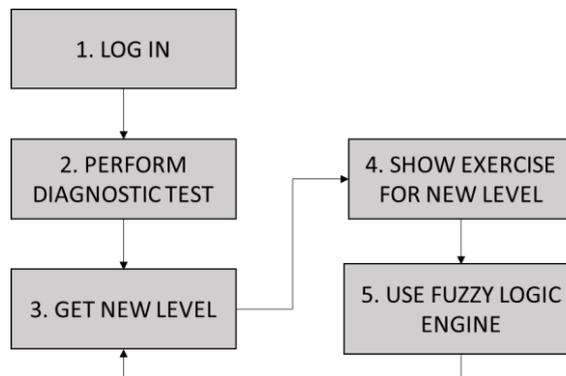


Fig. 5. Basic operation of the tutor.

#### 4 A Working Session with the ATS Java Zenzei

The ATS starts with a step of login for authentication and registration of users. In case of a new user the system proceeds to realize a diagnostic test, which consist in a set of

multiple choice questions. The result of the diagnostic test is the initial difficulty level of the student. The next step presents an exercise for programming in the Java language. As we explain before in section 1, a set of variables related to the solving of the problem define the new level of the student and the new exercise (see figure 5).

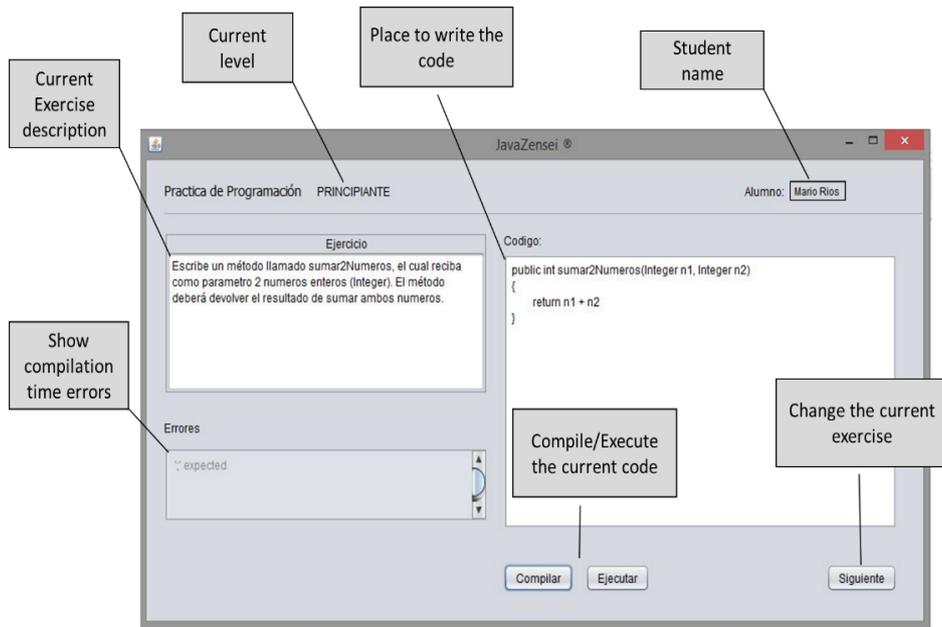


Fig. 6. Main screen of tutor JavaZenzei.

Once the student solve the exercise, a fuzzy inference engine defines the next exercise using a set of variables previously mentioned. The main interface of the Java Zenzei is shown in figure 6 with some brief explanation given in gray boxes.

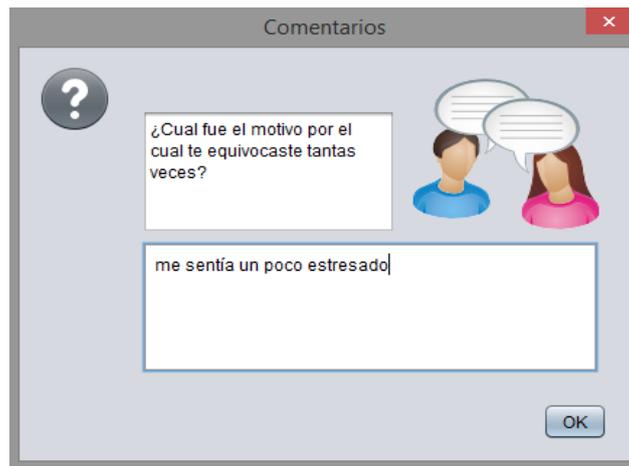


Fig. 7. Window to write comments on the exercises.

When a student ends an exercise a small windows is displayed asking to add comments related to that exercise (see figure 7).

## 5 Experiments and Results

We perform some initial test with graduate students from the Instituto Tecnológico de Culiacan. The exercises were designed for students with basic knowledge of Java language. In the test there were 9 students who finished the exercises in a short time and with few errors. That allowed students to move faster to more complex exercises. Table 1 shows the results obtained from the test.

Table 1. Result of the test.

Student	Gender	Diagnostic Level	Average Time	Failed Compilations	Average Emotion	Reached Level
E01	M	Basic	2.3 min	2	Neutral	Intermediate
E02	M	Basic	3.5 min	4	Surprise	Basic
E03	F	Basic	2.6 min	3	Neutral	Intermediate
E04	M	Advanced	2.1 min	0	Happy	Advanced
E05	M	Intermediate	2.3 min	2	Neutral	Advanced
E06	M	Intermediate	2.2 min	1	Happy	Advanced
E07	F	Basic	2.5 min	1	Neutral	Intermediate
E08	M	Intermediate	2.4 min	3	Surprise	Intermediate
E09	M	Intermediate	2.3 min	1	Neutral	Advanced

At table 1 we can see the student E06 for example, started at an intermediate level, had a short time solving the exercises, got just 1 compilation error, and the average emotion found was happy, so eventually the ITS promoted him to advanced level.

The rate of success of the emotion recognition was 80%. Whenever the student comments had more than 3 words that couldn't be found on the corpus SEL the emotion recognizer produce an incorrect results (incorrect emotion). The emotion recognizer achieves better results by adding those words in the corpus SEL. Finally whenever there was no match between the facial and textual emotion we choose the textual emotion as the correct one because most of the time the text emotion recognizer produce better results.

## 6 Conclusions

In this paper we presented a novel intelligent tutoring system for the support of learning the Java language. This system uses two emotion recognizers that work through text and facial expressions. We made different tests of the ATS with graduated students of computer science, producing the next results: the tutoring system based on facial expression is able to recognize emotions with a success rate of

80%. The emotion recognizer based on text dialogues achieved a success rate of 85%. The fuzzy system always produced the adequate exercises according to established parameters. For example, if the student was boring, had few errors, had few compilation and executions errors, and had a short time to solve the problem, the system increase the complexity of the exercises. On the other hand, if the student was not boring, had few errors and a greater time of solving the problem, the complexity of the next exercises remains the same. But if the student had too many errors and spent a lot of time solving the exercise, then the system decrease the complexity level of the next exercise.

Based on the obtained results we establish that combining both emotion recognizers (facial expression and text dialogues), add precision to the work of identifying the emotional state of a student. We found that the precision of the text-dialogue recognizer mostly depends on the number of words stored in the corpus Semantic.

As future work we expect to add more words to corpus Semantic, so we can increase the rate of success of the recognizer. Another upcoming work is to integrate more exercises with different rates of complexity with the goal of having a more complete ATS.

## References

1. Hernández, Y., Sucar, L. E., Arroyo-Figueroa, G.: Building an affective model for intelligent tutoring systems with base on teachers expertise. In: Proc. of MICAI, Springer, pp. 754–764 (2008)
2. Belavkin, R.: The role of emotion problem solving. In: C. Johnson (Ed.), Proceedings of the AISB '01 Symposium on emotion, cognition, and affective computing, Heslington, York, England, pp. 49–57, (2001)
3. Soleymani, M.: Multimodal Emotion Recognition in Response to Videos. IEEE transactions on affective computing, vol. 3, no. 2 (2012)
4. Bänziger, T., et al.: Emotion Recognition from Expressions in Face, Voice, and Body: The Multimodal Emotion Recognition Test (MERT). American Psychological Association (2009)
5. Sebe, N., Cohen, I., Huang. T.: Multimodal Emotion Recognition. WSPC (2014)
6. D’Mello, S., Graesser A.: Multimodal Semi-Automated Affect Detection from Conversational Cues, Gross Body Language and Facial Features. Springer Science+Business Media B.V (2014)
7. Graesser, A.: AutoTutor: an intelligent tutoring system with mixed initiative dialogue. IEEE Transactions on Education, Vol. 48 (2005)
8. Wu, C. H., Chuang, Z. J., Lin, Y. C.: Emotion recognition from text using semantic labels and separable mixture models. ACM transactions on Asian language information processing (TALIP), 5(2), 165–183 (2006)
9. Chaffar, S., Inkpen, D.: Using a heterogeneous dataset for emotion analysis in text. In Advances in Artificial Intelligence, pp. 62–67, Springer (2011)
10. D’Mello, S., Graesser, A.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 2, no. 4 (2012)

*Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Jorge García-Lizárraga, et al.*

11. Binali, H., Wu, C., Potdar, V.: Computational Approaches for Emotion Detection in Text. In: IEEE DEST (2010)