Session S3A

T-BOT and Q-BOT: A Couple of AIML-based Bots for Tutoring Courses and Evaluating Students

Fernando A. Mikic, Juan C. Burguillo, Daniel A. Rodríguez, Eduardo Rodríguez, and Martín Llamas
University of Vigo, mikic@det.uvigo.es, jrial@det.uvigo.es, darguez@det.uvigo.es, rodfer@enigma.det.uvigo.es, martin@det.uvigo.es

Abstract - Intelligent Tutoring Systems are computer programs that aim at providing personalized instruction to students. In recent years, conversational robots, usually known as chatterbots, become very popular in the Internet, and ALICE (Artificial Linguistic Internet Computer Entity) is probably the most popular one. ALICE brain is written in AIML (Artificial Intelligence Markup Language), an open XML language. We consider the combination of both approaches, i.e., the use of AIML-based bots for tutoring purposes in open e-Learning platforms like Claroline or Moodle. With that aim in mind, we have developed two different bots for helping the students during the learning process and for supporting the teaching activities of the professor. One of them is a tutor bot (T-Bot), and is able to analyse the requests made by the learners in written natural language and to provide adequate and domain specific answers orienting the student to the right course contents. The other one is an evaluation bot (Q-Bot), and is oriented to track and supervise the student progress by means of personalized questionnaires. Both bots have been already developed and integrated as user-friendly modules in Claroline and Moodle.

Index Terms - AIML, ALICE, chatterbots, Claroline, Intelligent Tutoring Systems, Moodle, Q-BOT, T-BOT.

INTRODUCTION

Intelligent Tutoring Systems (ITSs) [1] started to be developed in the 1980’s. They were designed as a knowledge-based, intelligent tutor, to guide the student in the process of learning; trying to emulate the way a human tutor guides the student in his/her learning path.

An intelligent tutor is a software system that uses Artificial Intelligence techniques to represent the knowledge and interacts with the students in order to teach them. In the 1990’s, with the advances of cognitive psychology and the new programming paradigms, ITSs have evolved from a mere instructional proposal to the design of environments for knowledge discovery and experimentation. In spite of their advances, ITSs have not been delivered to a general audience. This, in part, is due to the complexity implied in their design, which has limited their practical application [2]. Developing and implementing an ITS is a difficult task, since the required technology often implies most of the areas of Artificial Intelligence (AI): knowledge representation, diagnosis, cognitive modeling, qualitative processing, and causal modeling process. Besides, it is necessary to have a good knowledge of the domain or topic selected to be taught.

In the last decade, we also have witnessed the irruption of ALICE (Artificial Linguistic Internet Computer Entity) [3], one of the most ground-breaking projects in the field of Artificial Intelligence. ALICE is the project that produced the AIML (Artificial Intelligence Markup Language) language through which is possible to develop software chatterbots [4]. ALICE has won the “Loebner Prize in Artificial Intelligence Contest” [5] (based on the Turing test) several times.

An ALICE-like bot can be used as a tutor in an e-learning platform to provide tutoring and evaluating support. There are several chatterbots that act as tutors; some of the more representative ones are the following:

- TutorBot: Created by Orlando De Pietro, its main characteristic is the possibility to produce research into the diffused knowledge bases (e.g. Internet) through a direct interaction with a search engine [6].
- ELEKTRA: It was created at UFRGS (Federal University from Rio Grande do Sul) and its general objective is to help students in distance learning [7].
- BonoBot: A chatterbot with pedagogical agent characteristics developed to help ITS users [8].
- MEARA: Its main objective is to offer students several types of resources and materials, such as videos, figures, and illustrative examples [9].

In this paper, we will present two bots based on the AIML language for tutoring purposes, which will help students and tutors to improve their learning process. The main contributions of both bots are the following:

- The provision of adequate, domain specific, and personalized, information to the students.

Claroline and Moodle are e-learning platforms that help students in their learning tasks. They help learners by giving them access to course contents that these learners cannot personally attend, or simply by improving their learning process by adding multimedia resources to learning contents. Some of the features of these platforms are their ability:
• To run on the widest variety of software platforms, because they use PHP (hypertext preprocessor) and MySQL (database management system).
• To be easily installed, upgraded, learned, and modified.
• To be modified (due to their modularity and scalability): themes, activities, interface languages, database schemas, and course formats.
• To address interoperability issues (main educational standards are followed).
• To uphold the open software philosophy.

In spite of these advantages, there exist some important drawbacks that should not be overlooked. The lack of a tutor figure to pay specific attention to an individual student is one of those drawbacks. Here is where the bots discussed in this paper play an important role. These bots (T-BOT and Q-BOT) are dedicated to tutoring students, taking advantage of AI techniques and offering a kindly interface to the users.

There are several benefits that are obtained from using bots to help students in their learning process within an e-learning platform. For example, a bot is available to help students at any time of the day, any day of the week. A bot does not get bored or loses its patience due to the students' attitude, and it can attract and keep students’ attention because it supposes a technological innovation. Even to some degree, a bot can supply emotional intelligence to the learning process, making the student feel more comfortable than just surfing through the resources.

In following sections we will see how T-BOT and Q-BOT, a couple of AIML bots oriented to e-learning platforms, work.

T-BOT AND Q-BOT: COMMON ISSUES

T-BOT and Q-BOT are two virtual assistants designed to tutor and evaluate students into an e-learning platform. Both have been designed as PHP modules so they can be easily integrated into e-learning platforms, both as independent modules as well as integrated in only one module. These two e-learning bots were developed using the A.L.I.C.E. technology, and so their brains are compounded by a set of AIML files consisting of simple stimulus-response modules. They are based on Program E [12], the PHP implementation of the AIML interpreter, which provides a natural and faster way of integrating them inside a PHP module to be deployed into the corresponding platform.

These bots also use AJAX (Asynchronous JavaScript And XML) technology [13], that enables to make interactive applications or RIA (Rich Internet Applications). This technique enables our bots to maintain an asynchronous communication with the server in the background, and so, it is possible to make changes on the web page without to actualize it. This means a significant improvement of the interactivity, speed, and usability.

Lastly, we must remark that both bots support three languages: English, Spanish, and Galician (the bots select the language to use depending on the one chosen by the user at the platform).

On the other hand, referring to the system architecture at the highest level of abstraction, our system consists of an e-learning platform (Claroline or Moodle) with its own traces database and a chatterbot (T-BOT and/or Q-BOT) consisting of a Bot User Interface, an interpreter, and an AIML database. Figure 1 shows us the architecture with the main elements involved in the system.

The students are able to connect to the e-learning platform through the Internet and they will interact with the bot through the BUI (the Bot User Interface is a text box where the students can input their utterances). The bot gets input data from this BUI and searches into its source of knowledge appropriate content to reply; on the one hand the content provided by the e-learning platform through its modules (this information is directly supplied by the teacher of the course), and on the other hand a set of predefined AIML files. Moreover, the bot can access to student’s traces and questionnaire evaluations using the e-learning platform. Armed with this data, and using topics and keywords from the course itself, it influences the learner to concentrate on those didactic aspects that need further revision, and it supervises all the student activity providing feedback and encouraging students when necessary.

T-BOT FEATURES

As we stated before, T-BOT is an artificial intelligence entity designed to help the students through the learning
process. In order to achieve this, it embraces a virtual assistant guiding and tutoring students in their access to platform's resources. To achieve its goal, T-BOT is able to answer students' questions about different subjects using natural language.

To tell about T-BOT performance, we will consider two points of view: the tutor’s point of view and the students’ one.

I. Tutor's point of view

The tutor is the only person that can manage the bot, and he/she do this through the bot management module. This module offers a friendly interface for selecting the general AIML files to be loaded as well as to decide whether the bot is visible or not. Besides, it is possible to customize the bot giving it a name, hobbies, home town, birthday, etc. and a background image for the conversation window (see Figure 2). Taken all together, this makes the bot more human to interact with the students.

![CUSTOMIZATION OF THE BOT.](image)

The next step is to load the content course specific knowledge. In order to do that, T-BOT has an interface where teachers can select the resources from a structure of modules, choose several keywords to identify those resources, and create a brief abstract to be displayed within the bot answers (every resource will have a learning path). Having a learning path assures that knowledge perfectly reflects the structure given to the course by the teacher. Then, MySQL database tables include AIML-like entries where the keywords are the patterns and the learning paths ids are the templates. So, an association between the interpreter input-triggering pattern and the resources is created.

In addition, this module has an interface that follows the evaluation of the course given by the students. The information is shown in a table containing all the keywords associated with the course contents and the gathered statistics for these concepts. These statistics are:

- The number of times that the bot has given an answer to a specific concept (Frequency).
- The obtained score taking in account the votes of the learners, based on their satisfaction degree over the given answers by the bot (Rank).

All this information is an important feedback to realize how much a resource is referred and how good the given information by the bot is.

At last, we must remark that the tutor is able to talk to the bot in a similar manner to the student. This functionality helps the teacher to test how the bot works in a real environment (the bot interaction with the students will be addressed at the next section).

II. Students' point of view

The bot entity gives students a virtual tutor capable of solving their doubts as a real one. As has been stated above, students’ interaction with the T-BOT came across the Bot User Interface (BUI), which consists in a pop-up window with a text area reflecting the conversation and a text box to introduce new requests. So, the students are able to maintain a conversation with the bot through this BUI. This conversation can be a general conversation (thanks to a set of predefined AIML files that contain general information) or a specific one related to the content of a course (content given to the bot by the e-learning platform through its modules).

If the bot does not detect and input related to the content of a course, it answers to the students with an expression taken from its general knowledge base. Once the bot detects a reply from the student, where he/she has used a special open source.

- Online available.
- Versatile and sound.
- Supported by several languages (English, Spanish, Galician, Italian, and Catalan).
- Simply to install and make operational.
- Modular (which permits changes of the syntactic analyzer in future versions of the bot).

At this point, it is important to realize how T-BOT relates its knowledge to course content. Whenever a specific course content is going to be loaded into the database, the keywords of a resource are extracted, as well as the learning path. Having a learning path assures that knowledge perfectly reflects the structure given to the course by the teacher. Then, MySQL database tables include AIML-like entries where the keywords are the patterns and the learning paths ids are the templates. So, an association between the interpreter input-triggering pattern and the resources is created.

In addition, this module has an interface that follows the evaluation of the course given by the students. The information is shown in a table containing all the keywords associated with the course contents and the gathered statistics for these concepts. These statistics are:

- The number of times that the bot has given an answer to a specific concept (Frequency).
- The obtained score taking in account the votes of the learners, based on their satisfaction degree over the given answers by the bot (Rank).

All this information is an important feedback to realize how much a resource is referred and how good the given information by the bot is.

At last, we must remark that the tutor is able to talk to the bot in a similar manner to the student. This functionality helps the teacher to test how the bot works in a real environment (the bot interaction with the students will be addressed at the next section).

978-1-4244-1970-8/08/$25.00 ©2008 IEEE

38th ASEE/IEEE Frontiers in Education Conference

S3A-9
keyword (related to a learning resource of a course), the bot will recover the previously established association and will process the learning path id. All needed information will be found at the database tables, and then T-BOT will show an answer consisting of (see Figure 3):

- The resource's abstract.
- Extra information about the resource: It is a link to all the content of the course related to the concept that the student was asking for.
- Related information: A set of links to any type of information related to the resource that the bot has found.
- Scoring the answer: The bot offers to the student the possibility of ranking the given answer.

The Q-BOT then admits responses with a multiple choice format, true/false format, and short answers format. The type of the answers must be specified at the moment they are defined.

The interface also allows tests to be defined (see Figure 4). A test consists in several questions selected from those created. The teachers have the capability of specifying the order in which the questions will appear within a test, as well as the order in which the tests will be proposed to the students; to carry out their learning tasks. These questionnaires are personalized to the skills of the individual student. To do that, Q-Bot gathers information traced from within the platform database. The Q-Bot then analyzes these traces to determine the progress of the student: questionnaire marks, time needed to solve the questions, number of correct, incorrect, and not answered questions, etc. Afterwards, and using this information, the bot is able to adapt the difficulty of the follow-up questions to the student, based on their skill and expertise. In this way, Q-BOT provides teachers with an easy way to support student evaluation and to obtain direct feedback about student difficulties (analyzing their progress). In this way, teachers can concentrate on those concepts that require more expertise and real human interaction.
Another important fact about the questions is that a weight is given to each one shown after each evaluation, so it is possible to maintain and update a list of the most difficult questions (teachers can use this information, for example to edit those questions that they consider inappropriate).

Q-BOT also has an interface to create and edit clues. Clues will be assigned to a concrete question and must be ordered if there exist more than one. Clues will have an associated penalization so, if a student asks for help, a correct answer will not count the same as a correct answer given without help. The teacher can select how much a clue will penalize the scoring. As questions and answers, clues need an AIML-like entry to be added in the database of the bot.

Finally Q-BOT offers the possibility of keeping track of the statistics. The system generates several graphs to track the evolution of the student’s level and to show the current level of the questions. In order to do that Q-BOT uses an algorithm based in a simplified version of the IRT (Item Response Theory) [15]:

- Right answer: \( C = 0.2 \cdot (2^U - P) \)
- Wrong answer: \( C = 0.2 \cdot (2^P - U) \)

Where \( U \) is the user level and \( P \) is the question level. Both parameters are real numbers, when these are applying to the functions the closest integer is used. \( C \) is the amount in which the level of the user grows or decreases after he/she had answered a question. The value of \( C \) is set to a maximum of 2, even although it becomes greater than 2, to avoid an abrupt variation of the level in the case a user guesses a question. The levels will actualize every time a user answers a question.

II. Students’ point of view

The Q-BOT entity gives students a virtual evaluator capable of evaluating and monitoring their progress as a real one. Students’ interaction with the Q-BOT is made through a user interface consisting in a pop-up window with a text area and a text box.

When the student starts a conversation with the bot, it shows him/her a short introduction about its performance. At this moment the student can choose among the following options (see Figure 5):

- Ask for a test: The bot chooses the first one over all the tests the teacher had created that this student has not done.
- Ask for a personalized test: The student must choose the number of questions the test will include and the bot will create the test that fits this requisite.
- Ask for questions that do not belong to a test (free questions): The bot starts to ask questions and keeps on it until the student tells to it that he/she wants to stop.
- Start a general conversation: The bot will maintain a conversation with the student until he/she decides to choose one of the other above options. In this case, if the student spends time with a general conversation, the bot will remember him/her that he/she can to asking it for questions or tests.

At last, only to point out that the student can ask for a clue to answer a question, and that this fact will penalize his/her final score.

CONCLUSION

The main contribution of this paper is the use of artificial intelligence entities (AIML-based chatterbots) to enrich the learning process inside an e-learning platform. This is done through both tutoring and evaluating students for a course using natural language. A really important fact is that these bots can be easily integrated in open source e-learning platforms like Moodle and Claroline, offering a user friendly interface and the support of several languages.

T-Bot is a tool capable of answering students' requests (using natural language). It provides students a new way to look for information as well as guide them through the resources of the platform while keeping a coherent dialogue. Besides, feedback is allowed assessing the provided answers so further adaptation and enhancements can be performed by teachers.

Q-Bot has proved itself as a valuable supervising and evaluating tool. It allows students to observe their progress inside the course and evaluate the quality of the questions provided (always using natural language). Besides, level of the questions is always adapted to the student’s level. Teachers also can track student’s progress and re-adapt the level of the questions and tests if necessary.
ACKNOWLEDGMENT

This work has been funded by Ministerio de Educación y Ciencia through project “Servicios Adaptativos para E-learning basados en estándares” (TIN2007-68125-C02-02), and Consellería de Innovación e Industria de la Xunta de Galicia (Programa de Promoción General de la Investigación del Plan Gallego de IDIT) through project “E-BICS: E-learning – Bases de Integración e Coordinación sobre eStándares” (PGIDIT06PXIB32 2270PR).

REFERENCES


AUTHOR INFORMATION

Fernando A. Mikic received the M.Sc. degree in Engineering in 2001 from the University of Vigo, Spain. He is a Ph.D. candidate in Information Technologies at the University of Vigo, Spain. He is currently an assistant professor at the Department of Telematic Engineering at the same university. His research interests include intelligent agents, distributed artificial intelligence, e-learning, and telematics services. mikic@det.uvigo.es

Juan C. Burguillo received the M.Sc. degree in 1995 and the Ph.D. degree (cum laude) in 2001 from the University of Vigo, Spain; both in Telecommunication Engineering. He has been assistant professor during the period 1996-2001 at the ETSI of Telecommunications (Higher Technical School of Telecommunications Engineering) in the University of Vigo, Spain. He is currently an associate professor at the Department of Telematic Engineering at the same university. His research interests include intelligent agents, distributed artificial intelligence, distributed optimization, and telematic services. jurial@det.uvigo.es

Daniel A. Rodriguez received the M.Sc. degree in Engineering in 2004 from the University of Vigo, Spain. He is a Ph.D. candidate in Information Technologies at the University of Vigo, Spain. He is currently a researcher at the Information Technologies Group of the Department of Telematic Engineering at the same university. His research interests include ambient intelligence, grid computing, e-learning, and telematics services. darguez@det.uvigo.es

Eduardo Rodriguez received the M.Sc. degree in Engineering in 2007 from the University of Vigo, Spain. He is a Ph.D. candidate in Information Technologies at the University of Vigo, Spain. He is currently a full-time researcher at the Information Technologies Group of the Department of Telematic Engineering at the same university. His research interests include intelligent agents, multi-agent systems, e-learning, and telematics services. rodfer@enigma.det.uvigo.es

Martin Llamas received the Telecommunications Engineer degree (1986) and Doctor in Telecommunications degree (1994), both from the Technical University of Madrid, Spain. Since 1987 he has been lecturing at the ETSI of Telecommunications (Higher Technical School of Telecommunications Engineering) of the University of Vigo, Spain (where he was assistant head during the period 1994-1997). He is currently an associate professor of the Department of Telematic Engineering at the same university. His areas of interest are protocol engineering, e-learning, web engineering, and Internet based applications. Since December 1998 to October 2003 he has been the Head of ICT Area at the University of Vigo, Spain. He is a senior member of the IEEE. martin@det.uvigo.es