Journal of Universal Computer Science, vol. 15, no. 7 (2009), 1486-1495
submitted: 28/11/09, accepted: 28/3/09, appeared: 1/4/09 © J.UCS

TQ-Bot: An AIML-based Tutor and Evaluator Bot

Fernando A. Mikic Fonte
(University of Vigo, Vigo, Spain
mikic@det.uvigo.es)

Juan Carlos Burguillo Rial
(University of Vigo, Vigo, Spain
jrial@det.uvigo.es)

Martín Llamas Nistal
(University of Vigo, Vigo, Spain
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 have considered the combination of both approaches, i.e, the use of AIML-based bots for tutoring purposes in open e-Learning platforms such as Claroline or Moodle. With that aim in mind, we have developed a bot (chatterbot) for helping the students during their learning process and for supporting the activities of the teacher. This bot (TQ-Bot) 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. Besides, TQ-Bot is able to track and supervise the student progress by means of personalized questionnaires. This bot has been developed and integrated as user-friendly modules in Claroline and Moodle.

Keywords: AIML, ALICE, BDI, chatterbots, Claroline, Intelligent Tutoring Systems, Moodle, TQ-BOT
Categories: K.3.1, I.2.7

1 Introduction

Intelligent Tutoring Systems (ITSs) [Corbett, 97] 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 (AI) 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 [Sierra, 06]. Developing and implementing an ITS is a difficult task, since the required technology often implies most of the areas of AI: knowledge representation, diagnosis, cognitive modelling, qualitative processing, and causal modelling process. Besides, it is necessary to have a good knowledge of the domain or topic selected to be taught.

In the last decade, we have also witnessed the appearance of ALICE (Artificial Linguistic Internet Computer Entity) [ALICE, 95], 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 [Neves, 02]. ALICE has won the "Loebner Prize in Artificial Intelligence Contest" (based on the Turing test) several times.

AIML is a text file with a specific structure, which constitutes the knowledge base of a chatterbot. The "categories" are the fundamental knowledge basis, and they consist of at least two elements: the "pattern" and the "template". In general, the performance of AIML is based on a stimulus-response model, in which the stimulus (the user's input) corresponds with the "pattern", and the response (which the chatterbot will show to the user) will be its associated "template". All these actions, about looking for the adequate pattern and showing the related template, will be carried out by a data treatment engine, of which there are many versions (Program D, Program E, etc.).

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 most 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 [De Pietro, 06].
- ELEKTRA: It was created at UFRGS (Federal University from Rio Grande do Sul) and its general objective is to help students in distance learning [Leonhardt, 03].
- BonoBot: A chatterbot with pedagogical agent characteristics developed to help ITS users [Bianchi, 03].
- MEARA: Its main objective is to offer students several types of resources and materials, such as videos, figures, and illustrative examples [Leonhardt, 03b].

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

- Complete integration within open e-learning platforms such as Claroline [Claroline, 00] and Moodle [Moodle, 02].
- 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 managed.
- 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 our bot (chatterbot) plays an important role. TQ-Bot is dedicated to tutoring students, taking advantage of AI techniques and offering a kindly interface to the users.

The aim of TQ-Bot within an e-learning platform is to help students in their learning process in many ways. For example, our bot is available to help students at any time of the day, any day of the week. It 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, TQ-Bot can make the student feel more comfortable than just surfing through the resources.

In following sections we will see how TQ-Bot, an AIML bot oriented to e-learning platforms, works. We will present its general characteristics in section 2, and its features in section 3. At last, in section 4, we will finish the paper with conclusions and future work.

2 TQ-Bot General Characteristics

TQ-Bot is a virtual assistant designed to tutor and evaluate students (it was designed as a PHP module, so it can be easily integrated into e-learning platforms). This bot was developed using the A.L.I.C.E. technology, and so their brain are compounded by a set of AIML files consisting of simple stimulus-response modules. It is based on Program E [Program E, 02], the PHP implementation of the AIML interpreter, which provides a natural and faster way of integrating it inside a PHP module to be deployed into the corresponding platform.

This bot also uses AJAX (Asynchronous JavaScript And XML) technology [Asleson, 06], that enables to make interactive applications or RIA (Rich Internet Applications). This technique enables our bot to maintain an asynchronous communication with the server in the background, and so, it is possible to make changes on the web page without reloading it. This means a significant improvement of the interactivity, speed, and usability

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 (TQ-Bot) which is composed of a Bot User Interface (BUI), 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 knowledge source 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 through a set of predefined AIML files. Moreover, the bot can access student's traces and questionnaire evaluations using the e-learning platform. According to these data, and using topics and keywords from the course itself, it influences the learner to concentrate on those didactic aspects that need further revision.

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

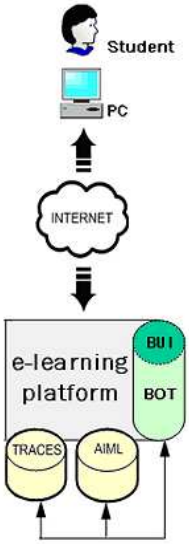

Figure 1: Main architecture of the e-learning platform with TQ-Bot embedded

3 TQ-Bot Features

To tell about TQ-Bot performance, we will consider two points of view: the students' point of view and the tutor's one.

3.1 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 TQ-Bot is handled across the 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, which 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 any 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 keyword (related to a learning resource of a course), the bot will recover the previously established association and will process the learning path. All needed information will be found at the database tables, and TQ-Bot will show an answer consisting of (see Figure 2):

- The resource's abstract.
- Extra information about the resource: 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.

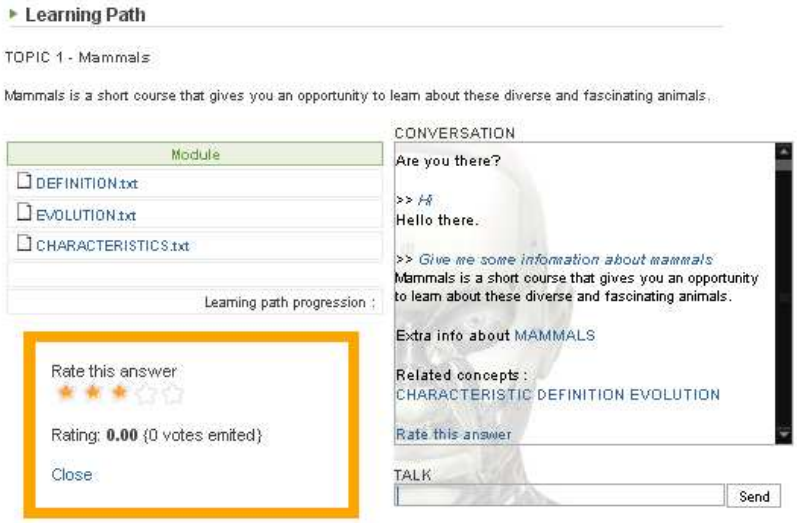

Figure 2: Answer of the TQ-Bot

TQ-Bot also gives students a virtual evaluator capable of evaluating and monitoring their progress as a real one. While a student is talking to the bot, he/she can request several activities (see Figure 3):

- Ask for a test: The bot chooses the first one over all the tests the teacher had created that this student has not done yet.
- 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 wants to stop.

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.

Figure 3: Talking to the bot

3.2 Tutor's Point of View

3.2.1 Installation

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 4). Taken all together, this makes the bot more human to interact with the students.

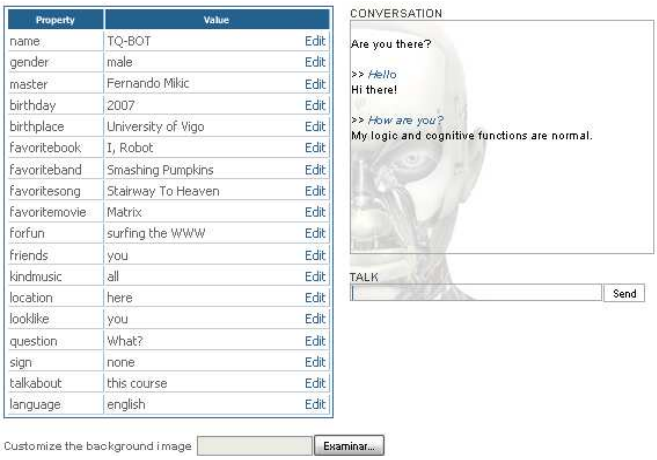

Figure 4: Customization of the bot

The next step is to load the content course specific knowledge. In order to do that, TQ-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.

Teachers can introduce themselves the keywords, but initially they will be recommended by an automatic syntactic analyzer. The first time that a specific knowledge is associated to a bot, a syntactic analysis is performed and the keywords are shown, i.e. the syntactic analyzer is used to obtain a recommendation of keywords/concepts related to the course content. Here, FreeLing 1.5 [Carreras, 04] is employed, an open source linguistic analyzer.

The reasons to choose this analyzer are the following:

- Open source.
- Online available.
- Versatile and sound.
- Supported by several languages (English, Spanish, Galician, Italian, and Catalan).
- Simple 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 TQ-Bot relates its knowledge to course content. Whenever 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 will 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 into 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.

3.2.2 Students' evaluation

Another important characteristic of TQ-Bot is the possibility of offering teachers a simple and intuitive user interface to introduce questions assigned to the platform's resources (obviously, every question must have at least one associated answer). Whenever a question (and its answer) is assigned to a resource, it must be added to the AIML database of the bot (in order to do this, some functions were developed).

TQ-Bot 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 5), which consist in several questions selected from those created. The parts of Figure 5 are detailed below:

- Tests in this course: It is a list of tests which belong to the present course. From here it is possible to change the order of each test, to access it, and to delete it.
- Add a new test: To add a new test to the present course.
- Question bank: To access several actions which can be performed over the questions: creation, editing, sorting, deleting, etc.
- Questions in test "Test Name": It shows the questions belong to the test called "Test Name". From here it is possible to change the order of each question within the test and to delete it.

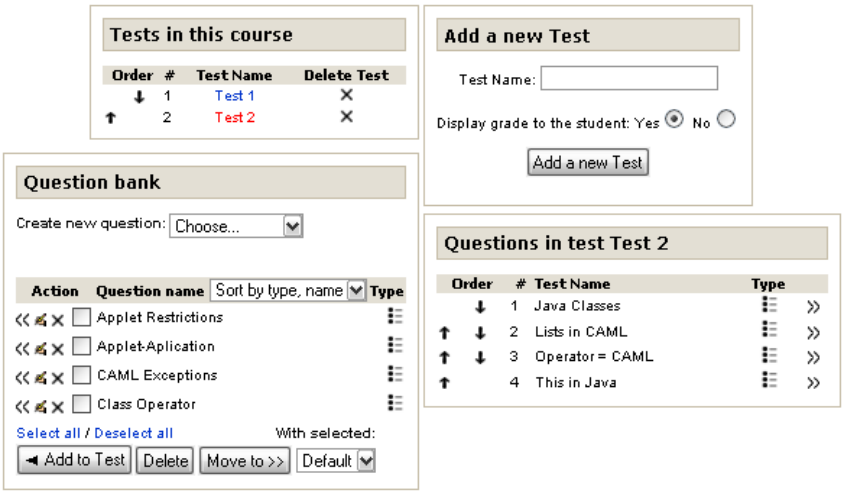

Figure 5: Test management

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. Besides, these questionnaires are personalized to the skills of each individual student. To do that, TQ-Bot gathers information traced from within the platform database, and then, it 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, TQ-Bot provides teachers with an easy way to support student evaluation and to obtain direct feedback about student difficulties (analyzing their progress). So, teachers can concentrate on those concepts that require more expertise and real human attention/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).

TQ-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, related to questions and answers, TQ-Bot offers the possibility of keeping track of 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 TQ-Bot uses an algorithm based in a simplified version of the IRT (Item Response Theory) [Hambleton, 91]:

- Right answer: $C=0,2*(2^{(P-U)})$.
- Wrong answer: $C=0,2*(2^{(U-P)})$.

Where U is the user level and P is the question level (both parameters are real numbers, and when these are applying to the functions the closest integer is used). The question level shows how difficult a question is (the more it is wrong answered more difficult it is). The user level reflects the knowledge of the student, and it grows or decreases after he/she had answered a question in an amount of C. 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 are updated every time a user answers a question.

4 Conclusions

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

TQ-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, and so, further adaptation and enhancements can be performed by teachers.

This bot can also be used 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 (besides, the level of the questions is always adapted to each student's level). At last, teachers also can track student's progress and re-adapt the level of the questions and tests if necessary.

TQ-Bot was initially developed to help students in their learning tasks within Claroline and Moodle. Nevertheless, it is being evolved to be part of a new e-learning platform, which we are nowadays developing, called INES (Intelligent Educational System).

Acknowledgements

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

[ALICE, 95] ALICE Artificial Intelligence Foundation, 1995, http://www.alicebot.org/

[Asleson, 06] Asleson, R., Schutta, N. T.: Foundations of AJAX, 2006.

[Bianchi, 03] Bianchi Sganderla, R., Nice Ferrari, D., Geyer, C. F. R.: BonoBOT: Um Chatterbot para Interaçao com Usuários em um Sistema Tutor Inteligente, XIV Simpósio Brasileiro de Informática na Educação – SBIE 2003, Rio de Janeiro, Brazil, pp. 463-472, 2003.

[Carreras, 04] Carreras, X., Chao, I., Padró, L., Padró, M.: FreeLing: An Open-Source Suite of Language Analyzers, 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal, pp. 239-242, 2004.

[Claroline, 00] Claroline, 2000, http://www.claroline.net

[Corbett, 97] Corbett, A. T., Koedinger, K. R., Anderson, J. R.: Intelligent Tutoring Systems, Handbook of Human-Computer Interaction, Completely Revised Edition in M. Helander, T. K. Landauer, P. Prabhu (Eds), Elsevier Science B. V., c. 37, pp. 849-874, 1997.

[De Pietro, 06] De Pietro, O., Carmelo, P., De Rose, M., Frontera, G.: An Intelligent Agent and an Adaptive Search Engine to support tutoring activities on-line, Journal of e-learning and knowledge society, v. 1, n. 1, pp. 97-110, 2006.

[Hambleton, 91] Hambleton, R. K., Swaminathan, H., Rogers, H. J.: Fundamentals of Item Response Theory, Sage Press, Newbury Park, CA, 1991.

[Leonhardt, 03] Leonhardt, M. D., Dorneles de Castro, D., Rockenbach Tarouco, L. M.: ELEKTRA: Inteligência Artificial na Educação a Distância de Jovens e Adultos, Congresso de Educação a Distância Mercosul, pp. 165-170, 2003.

[Leonhardt, 03b] Leonhardt, M. D., Neisse, R., Rockenbach Tarouco, L. M.: MEARA: Um Chatterbot Temático para Uso em Ambiente Educacional, XIV Simpósio Brasileiro de Informática na Educação – SBIE 2003, Rio de Janeiro, Brazil, pp. 85-92, 2003.

[Moodle, 02] Moodle, 2002, http://moodle.org

[Neves, 02] Neves, A. M. M., Diniz, I., Barros, F. A.: Natural Language Communication via AIML Plus Chatterbots, V Symposium on Human Factors in Computers Systems (IHC 2002), Fortaleza – CE, p. 386, 2002.

[Program E, 02] Program E, 2002, http://sourceforge.net/projects/programe/

[Sierra, 06] Sierra, E., Martínez, R., Cataldi, Z.: Towards a Methodology for the Design of Intelligent Tutoring Systems, Research in Computing Science Journal, n. 20, pp. 181-189, 2006.