In this paper we present and discuss the results of the evaluation of an Intelligent Computer Assisted Language Learning (ICALL) system that operates over the Web. In particular, we aimed at evaluating the system along three dimensions: a) the effect of the intelligent features of the system on the learning outcome of students, b) the system’s ability to provide individualized support to students that leads to more effective use of the system and c) the general usability and friendliness of the ICALL. To achieve this, we conducted an empirical study, where we compared the intelligent system with a non-intelligent version of it. The results of the study revealed that the students of the Web-based ICALL had gained more knowledge of the domain and had been able to interact with the system more effectively as compared to the students that had used the non-intelligent version of the system. However, the students of the intelligent version of the system found it more difficult and they needed more time to get acquainted with the system in comparison to the students of the non-intelligent system.

Keywords: Adaptive web-based education; Evaluation; Usability; Intelligent tutoring systems; Student modeling.

1. Introduction

The increasing popularity of the WWW and the Internet has determined a trend towards the development of Web-based educational applications. Web-based educational systems aim at reaching learners of varying backgrounds, in situations where no teacher may be available to help the students. To accommodate such diverse audience, these systems should be as individualized as possible. Therefore, to achieve personalization over the WWW, a lot of research effort has been put on the development of intelligent Web-based educational systems. Indeed, in the field of Computers in Education, the technology of Intelligent Tutoring Systems (ITS) has been able to provide a high degree of individualization of educational applications to the needs of the particular students who
are using them. To a large extent this is achieved through the student modeling component that has traditionally been one of the most important parts of an ITS.

Intelligent Web-based educational systems, in common with standalone ITSs, aim at providing individualized support to students based on the information held in the model of each student. It has been a common practice for Intelligent Web-based educational applications to employ adaptive hypermedia techniques. These techniques are particularly useful for the Web, where the problem of navigation of students may render the Web-based educational applications difficult to use. Popular adaptive techniques include adaptive navigation support and adaptive presentation.

The main goal of intelligent instruction and support is to lead to an increased learning outcome. However, the achievement of this goal cannot be foreseen. Therefore, in order to measure the effectiveness, the efficiency and the usability of such systems, they have to be empirically evaluated using real students. Although the necessity of evaluation of intelligent systems has been widely acknowledged, there are few empirical evaluation studies reported in the literature of Web-based educational research. Furthermore, the studies that have been conducted so far, in most of the cases, are concerned with revealing the effectiveness of a single intelligent feature of the Web-based educational system, for example adaptive navigation support. However, there are other important intelligent features, such as error diagnosis, which have been overlooked in evaluations. In view of the above, there is an obvious need for further research in the direction of evaluating the ability of intelligent educational systems to promote students’ learning, as it has been acknowledged by many researchers.

In this paper we describe an experiment that was conducted in order to empirically evaluate a Web-based Intelligent Computer Assisted Language Learning (ICALL) system. The aim of the experiment presented in this paper was to reveal the effect, if any, of the intelligent features of the system on the students’ learning and on their usage of the system. Furthermore, we also wanted to evaluate the system in terms of its usability and friendliness. To achieve this, we compared the ICALL system with a second version of it that did not make use of the intelligent features (CALL system). The evaluation was based on both quantitative (pre- vs. post-test comparison, log-files analysis) and qualitative (questionnaires) evaluation methods. Evaluating intelligent systems by comparing them with other non-intelligent versions of the same systems is a common practice. A problem of such kind of evaluations is that the non-intelligent versions may not be optimally designed. In order to overcome this problem the non-intelligent version of our system was carefully designed. In particular, instead of disabling the intelligent features from the existing ICALL system, the CALL version of the system was developed from scratch using the same teaching material as the one used in the ICALL version of the system. This particular choice was made in order to avoid malfunctioning of the CALL system due to errors that could occur in the process of removing the intelligent features from the ICALL system. Therefore, the two versions of the system that were evaluated offered the same course material to students and presented them with the same exercises to solve.

The results of the study discussed in this paper provide evidence that the students who used the Web-based ICALL gained significantly more knowledge of the domain and had been able to interact with the system more effectively as compared to the students that used the CALL system. In accordance with the results of the analysis of the objective data, the students’ answers to the questionnaire that aimed at evaluating the usability and friendliness of the system showed that the students were significantly more satisfied from
the individualized support and instruction provided by the intelligent version of the
system as compared with the one-size-fits-all tutoring of the CALL system. Furthermore,
both groups of students rated their enjoyment of the tutor on a similar scale. However, the
students of the ICALL rated it as harder to use and needed more time to get familiar to
using the system in comparison with the students of the non-intelligent system.

2. Overview of the Web-based ICALL

The Web-based ICALL that was evaluated is a system that aims at teaching the domain
of the passive voice of the English language and is called Web-Passive Voice Tutor
(Web-PVT). Web-PVT incorporates techniques from Intelligent Tutoring Systems and
Adaptive Hypermedia to tailor instruction and feedback to each individual student.
Individualization in Web-PVT is based on two models: the domain model (representing
knowledge about the domain of the passive voice of the English language) and the
student model (representing knowledge about the individual student). In the following
subsections we will describe the domain and student models and present how these two
models are used in order to provide personalized instruction.

2.1. Representation of the domain knowledge

To enable communication between system and learner at content level, the domain model
of the system has to be adequate with respect to inferences and relations of domain
entities with the mental domain of a human expert. In this sense, the domain knowledge
of Web-PVT is represented in a conceptual network that depicts the interrelations
between the several grammatical concepts of the domain of the passive voice of the
English language. Representing the domain knowledge in a structured way ensures that
the system “knows” the dependencies between concepts, and uses this knowledge to
individualise instruction and provide adaptive problem solving support and feedback to
errors.

Similarly with KNOME, the grammatical concepts are grouped in categories based
on their level of difficulty. These categories include simple, mundane and complex
concepts. In Web-PVT the identification of the difficulty level categories was a result of
an empirical study conducted before the system’s development. Furthermore, each node
in the domain knowledge represents a certain category of concept, which may be further
divided into smaller sub-concepts. There are three kinds of link between nodes: part-of,
is-a, and prerequisite. A part-of relation points from a more general to a more specific
concept, which is one of its parts. For example, the “verb tense conversion” concept is a
part of the mundane concepts. An is-a relation, points from an instance of a concept to the
concept. For example, there is an is-a relation between the several verb tense forms and
the “verb tenses” concept. A precondition relation points from a concept to another,
which is its prerequisite. For example, in order to master the simple past tense, one
should know how to form irregular verbs. Finally, hypermedia pages and exercise
templates examining the knowledge that must be acquired by studying a specific concept
are associated with this concept; these associations are also part of the domain
knowledge.
2.2. Student modeling in Web-PVT

An important component in the architectures of both Web-based ITSs and adaptive Web-based educational hypermedia systems is the student modeling component. Indeed, the student modeling module is the part of a Web-based educational application that is responsible for acquiring and representing the necessary information about each student. More specifically, the student modeling module performs two main functions:17

- Initializes the student model when a new student logs on the ITS for the first time.
- Updates the student model based on the student’s interaction with the system.

In the case of Web-PVT, the model of each student is represented using a set of feature vectors. The first vector is responsible for representing information acquired by the student using an interview and a preliminary test presented in her/his first interaction with the system. These characteristics include the name of the student, the knowledge level stereotype category that s/he belongs to, an estimation of how careful the student is while solving exercises, her/his mother tongue, as well as other languages that the student already knows. The second student model vector on the other hand is directly related to the domain knowledge of Web-PVT. This vector represents the system’s estimations about the degree of knowledge and error proneness of the student for each concept in the domain knowledge. In particular, for each one of the forty-two concepts that are contained in the domain knowledge of Web-PVT there are two feature-value pairs related to it in the student model. The first pair represents an estimation of the student’s degree of knowledge concerning this particular concept, whereas the second represents an estimation of the student’s proneness to make mistakes while using this concept.

The initialization of the model of a new student in Web-PVT is based on the Initializing Student Models (ISM) framework, a domain independent methodology for initializing student models in Web-based ITSs.18 The ISM framework uses a novel combination of stereotypes and the distance weighted k-nearest neighbor algorithm. In particular, stereotypes are used to make initial assumptions about the student’s knowledge level of the domain being taught. The distance weighted k-nearest neighbor algorithm then is used to set initial values to all aspects of the student model. This is done based on the student’s similarity with other students that belong to the same stereotype category. However, these students have already used Web-PVT for some time so that the system has been able to construct their individual student models that have been inferred from extensive observations of their behavior. The similarity between students is estimated taking into account the characteristics students’ mother tongue, their prior knowledge of other languages and their degree of carefulness when solving exercises.19 All the above student attributes are acquired based on the student’s answers to a questionnaire that is posed when s/he uses Web-PVT for the first time.

After having constructed an initial model of a particular student, Web-PVT updates this model based on the actual behavior of the student while interacting with the system. The initial values of the student model are updated both if they seem not to stand for the particular student (e.g. if the student makes many mistakes in exercises concerning a concept that is considered as known to her/him) and if the knowledge of the student evolves as s/he learns with the system. This is done based on the student’s studying of theory pages and her/his exercise solving performance. In particular, the characteristic of the student model that represents the knowledge level of the student in a domain concept
is increased when the student visits the theory page that is associated with this particular concept, or solves correctly an exercise that tests this concept. Furthermore, the characteristic of the student model that represents a student’s proneness to make mistakes concerning a specific concept is calculated based on the student’s erroneous answers to exercises that are associated with this concept. Finally, the degree of carefulness of the student when s/he solves exercises is estimated based on the number of accidental slips and typographic errors (e.g. anagrams, omissions of a letter in a word, etc.) that the student has made in her/his answers to exercises.

2.3. Personalized tutoring and support

Based on the information contained in the student model the system provides intelligent, personalized tutoring and support to the student. In particular, based on the information that concerns the knowledge level of the student in each concept of the domain knowledge, the system provides individualized support when s/he navigates through the course material. Web-PVT uses a combination of two link adaptation techniques to help the student while navigating through the structured theory hyperdocument; namely adaptive link annotation and direct guidance. The idea of adaptive annotation is to augment links with some form of comments, which can tell the student more about the current state of the nodes behind the annotated links. The annotation of the links in Web-PVT is provided by using different icons and font types that inform the student whether s/he is ready to visit the corresponding theory part or if the studying of a page is unnecessary due to the fact that the student has already mastered the concept that is associated with this page. With the direct guidance technique, on the other hand, a “next” or “continue” button is shown to the student. This button leads the student to the particular theory page that the system considers as the most appropriate for the student to visit. By using a combination of the above techniques, Web-PVT allows the student to select between the more restrictive help provided by the direct guidance technique and the help provided by the annotation of the links that assists her/him to decide which page s/he wishes to read. An example of the adaptive annotation of links and “Next” button in the table of contents of the theory hyperdocument is illustrated in Figure 1.

Fig. 1. Personalized table of contents of the theory hyperdocument.
Moreover, apart from being consulted in order to provide individualized support to students when they navigate through the course material, the student model is utilized in Web-PVT before presenting the student with new exercises to solve. There are three types of exercise supported in Web-PVT. Each type of exercise has a different difficulty level. The easiest exercises are the multiple choice exercises. The next level of difficulty is assigned to exercises where the student is asked to type a word to complete a passive sentence. Finally, the most difficult exercises are those where the student is asked to transform a sentence from the active to the passive voice and vice versa. Web-PVT has the ability to construct dynamically new exercises for the student to solve. This means that it incorporates syntactic and semantic knowledge concerning the words that may constitute a meaningful sentence. Hence, the system is able to show a wide range of new exercises to students.

In order to select the next exercise to present to the student, Web-PVT consults the individual student model. In particular, it uses the information that represents the knowledge level of the student in each domain concept. First, it looks for domain concepts related to parts of the theory that the student has already read. Then, among them, it looks for concepts that the student faces difficulty in using when solving exercises. The tutor then selects an exercise that evaluates as many of those concepts as possible. If there is no such concept, the system selects a concept for which the student has the greatest proneness to make mistakes while using it in exercises and presents an exercise testing this concept. Furthermore, the type of the exercise that is presented to the student (multiple choice, fill in the blank or rewrite the sentence exercise) is decided based on the degree of knowledge of the student in the concepts that are to be evaluated by the exercise. The greater the degree of knowledge of the student, the more difficult the type of exercise that is presented to her/him. In this way, we ensure that students are not always asked to solve either too easy or too difficult exercises.

Finally, in cases of error when the student solves exercises, Web-PVT uses the information recorded in the model of the particular student and a library of the most common mistakes that students make when using the passive voice of the English language in order to perform error diagnosis and advice generation. The library of the most common mistakes was created based on an empirical study that involved human teachers and students and which was conducted prior to the development of Web-PVT. Based on tests set to students, common mistakes that students frequently made while learning the passive voice of the English language were identified. Then human teachers classified common mistakes into nine broad categories of error that may be recognized by the system. These errors are associated with the conversion of passive into active voice and vice versa and also with certain prerequisite grammatical concepts, such as irregular verbs.

In multiple choice exercises, error diagnosis is simple. For each one of the erroneous alternatives, there is an associated misconception. The system consults the student model in order to determine how careful a student is while solving exercises, before presenting to the student the error message that is associated with the selection s/he has made. In case the student is considered “careless” when solving exercises, the system consults the model of this particular student, to find out whether the student seems to know the concepts that this question evaluates. If the student seems to have mastered these concepts, Web-PVT infers that the student's mistake was due to her/his carelessness and not due to some more serious reason. Else, the system presents to the student the error message that corresponds to the erroneous selection of the student. However, there are
cases when the student may be both careless and lacks the knowledge of the concepts that are evaluated by an exercise. In such cases, Web-PVT assumes that the student has made the mistake due to her/his lack of knowledge.

In the other two types of exercise, where the student is required to enter free text, error diagnosis becomes more sophisticated. While performing analysis of the student’s answer, the system ignores trivial typographic errors such as the absence of a fullstop at the end of the sentence, absence of any space between words, redundant spaces or commas, anagrams of a word, etc. However, these errors are recorded by the system in order to provide information concerning the degree of carefulness of the student when solving exercises. If the student has made an error other than trivial typographic errors, then the system performs error diagnosis taking into account information that has been collected about the specific student who has made the mistake as well as common students’ mistakes. In some cases a mistake of a student may be attributed to more than one categories of error. In cases like this the system takes into account the individual features of the student, that have been recorded in previous interactions, in order to resolve the ambiguity and formulate the kind of advice to give to her/him. Finally, the student’s level of knowledge is used in order to determine the amount of help to be provided to the student when s/he is solving exercises. The less proficient a student is the more help s/he receives.

3. Methods of the Evaluation

3.1. Aims and scope of the evaluation methods

Educational software has to be educationally beneficial to students otherwise it cannot be considered successful. This means that it has to improve students’ learning which is a difficult cognitive process. As a result, educational software is a special case of software that has to be evaluated taking into account the cognitive processes of students. Therefore, it is important to find out if students’ learning has in fact taken place, whether it has in fact been facilitated by the software’s abilities and whether it has not been hindered by possible complexities of the user interface.

In the case of intelligent educational software, such as Web-PVT, there has to be additional focus on the evaluation of the intelligent features of the software. The existence of intelligence in educational software usually means that there is a student modeling component. As Chin points out, adding a user model to any software system will most likely make it more complex, less predictable and more buggy; consequently, it is a very reasonable question to ask whether or not the user model will actually improve the system.

In view of the above, one of the primary aims of the evaluation of Web-PVT was to find out the extent of its effectiveness in students’ learning. The effectiveness of Web-PVT was measured by setting pre-tests and post-tests to students who used Web-PVT. The system would be considered successful if the students’ post-test achievement scores were significantly higher than the pre-test achievement scores. However, the fact that Web-PVT incorporated intelligence imposed the existence of more evaluation tests, which would show whether the intelligent features were worth the effort. Therefore we also created a non-intelligent version, which was also evaluated in terms of its effectiveness in students’ learning using pre and post-tests. Then the performance of
students of the two versions was compared. The experiments were performed “blind”. According to Chin, in the blind experiment, participants do not know if the software that they use has a user model and so is “supposed to be better”.

Furthermore, in order to achieve a more analytical evaluation of the intelligence of Web-PVT, we evaluated the main intelligent and adaptive facilities of Web-PVT separately. The main intelligent and adaptive facilities of Web-PVT were the adaptive navigation support as well as the error diagnosis and adaptivity of the system when students solved exercises. Both these facilities were based on the student modeling component of Web-PVT. In particular, adaptive navigation support is closely related to the sequence of pages that a student may visit and read. Therefore, the students’ actions that concerned their visits of pages were recorded and examined for the purposes of the evaluation. Similarly, the students’ actions that were related to the effect that the error diagnosis had on students’ behavior were also recorded and examined for the purposes of the evaluation. Then the students’ actions who used Web-PVT were compared to the students’ actions who had used the non-intelligent version of CALL.

Finally, one important factor affecting the effectiveness of educational software is the usability and friendliness of the system. Indeed, if the user interface formalities are difficult for a student to learn and use, then this poses an extra cognitive load on the student who is already loaded with the difficult cognitive process of learning the material to be taught. Therefore, Web-PVT was evaluated in terms of its usability and friendliness and was compared to the non-intelligent version of CALL. This evaluation was mainly conducted based on the students’ answers to questions about whether they liked their interaction with the system they had used.

3.2. Experiment set-up

As mentioned above, we compared the Web-based ICALL with a CALL version of it. The non-intelligent system contained the same information and had similar functionality with the ICALL system. However, it did not have a student modeler. Therefore, it was not capable of individualizing instruction and feedback to the needs of each student. More specifically, when studying theory or selecting an exercise to solve in the CALL, the student was just presented with all the available topics or exercises and was asked to select the one s/he wanted to visit. A next button was also present in the table of contents and in each one of the theory pages that could lead the student to the next topic to be studied in a linear order. In addition, the error diagnosis of the CALL system was not capable of performing ambiguity resolution. Therefore, in cases when the error of a student could be attributed to more than one categories of error, the system could not decide which was the most probable cause. Finally, all students of the CALL system were provided with the same amount of help when they made errors, which was a plain indication that they had made an error. This particular approach was very common in early Computer Assisted Learning systems. Furthermore, in this way, the non-intelligent tutor avoided showing to the student with many possible error codes when her/his error could correspond to more than one code.

The participants of the experiment were 102 students of the fifth and sixth grade of an elementary school. The students had varying background knowledge of the domain of the passive voice of the English language. However, all participants had similar training and experience on computers, the WWW and hypermedia. The students were randomly divided into two groups. The first group (ICALL group) that consisted of 51 students,
worked with the intelligent version of the system. The rest 51 students formed the control group of the experiment (CALL group) and were asked to work with a version of the system that did not make use of the student model-based individualization capabilities.

The experiment was designed to be conducted in a real-world teaching and learning context. Therefore, the students of each group were asked to interact with the respective version of Web-PVT within the context of their English courses over a two-week period. In particular, students were involved in the experiment one hour each day (session), for four days within the two weeks. In their first session, students of both groups were asked to register to the respective version of the system that they were going to use and then to work on a computer-based pre-test. The pre-test consisted of ten representative questions of varying levels of difficulty that were related to essential grammatical concepts being taught by Web-PVT. The students’ performance on the pre-test was recorded by the system in a personal log-file. Furthermore, for the ICALL group, the performance of each student on the pre-test was also used by the system to categorize students in a stereotype concerning their knowledge level. Then, students were introduced to each version of the system, by a brief presentation of its capabilities and guidelines for its usage.

In the two subsequent sessions the students of each group worked freely with the respective version of Web-PVT. While working with the system, the actions of each student were recorded in her/his personal log-file, using the method of computer logging. The main advantage of computer logging is that the system can automatically and continuously collect objective data for further analysis and interpretation without interfering with users during their interactions with the system. In particular, the personal log-file contained information concerning the time the student had spent studying each theory page visited, the number of exercises attempted, the number of the student’s requests to see the solution to an exercise, and the number of correct student’s solutions to exercises after the system had provided advice to the student’s errors.

Finally, in the last session, students of both groups were asked to take a computer-based post-test and to complete a questionnaire using paper and pencil. The post-test consisted again of ten representative questions of similar level of difficulty to those of the pre-test. As was the case in the pre-test, students’ performance on the post-test was recorded in their personal log-files. The questionnaire that was posed to the students in their last session concerned their experience from interacting with the system. The participants were asked to rank their perception on various issues that concerned the usability and friendliness of the system that they used on a Likert scale with five responses. The completed questionnaires were then gathered in order to be analyzed.

4. Results of the Evaluation

4.1. Results of the objective data analysis

The students’ pre- and post-test performance as well as their actions during their interaction with the respective system were compared using t-test statistics, with the alpha value set to 0.05. The results of the analysis of the objective data that were stored in the log-files of students are summarized in Table 1.
Table 1. Results of the analysis of the students’ log files.

<table>
<thead>
<tr>
<th>Variable</th>
<th>ICALL Group(n=51)</th>
<th>CALL Group(n=51)</th>
<th>t(df); p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-test score (between 0 and 10)</td>
<td>Mean Value</td>
<td>Standard Deviation</td>
<td>Mean Value</td>
</tr>
<tr>
<td>Post-test score (between 0 and 10)</td>
<td>6.04</td>
<td>1.48</td>
<td>6.12</td>
</tr>
<tr>
<td>Percentage of theory page visited for less than half a minute</td>
<td>18.02%</td>
<td>12.49%</td>
<td>36.31%</td>
</tr>
<tr>
<td>Percentage of requests for the solution to exercises</td>
<td>24.61%</td>
<td>8.37%</td>
<td>39.11%</td>
</tr>
<tr>
<td>Percentage of correct solutions based on the system’s feedback</td>
<td>45.39%</td>
<td>11.77%</td>
<td>26.46%</td>
</tr>
</tbody>
</table>

In particular, in order to investigate the effect of the intelligent features of Web-PVT on the students’ learning outcome, we compared the scores of the students of the ICALL group versus those of the CALL group on the pre- and the post-test. An initial analysis of the performance of the two groups on the pre-test indicated no significant difference \( t(100)=-0.24, \ p=0.405 \). This showed that the students of the two groups had similar prior knowledge of the domain of the passive voice of the English language. Then, after the students had interacted with the tutoring systems, in both experimental conditions there was an improvement on the students’ performance on the post-test as compared to the pre-test. However, the students of the ICALL group scored significantly higher in the post-test as compared to the students of the CALL group \( t(100)=2.27, \ p=0.013 \). This meant that Web-PVT had achieved educational benefits for the students to a higher extent than the non-intelligent version of it.

Furthermore, the effect that the student model-based adaptation had in the students’ usage of the system was measured based on a set of dependent variables. In particular, we wanted to evaluate the capabilities of Web-PVT to provide adaptive navigation support to students when they studied theory. For this purpose, we counted the pages of the theory hyperdocument that each student had visited during her/his interaction and we recorded the time that s/he had spent to read each of these pages. In cases, when the student had spent a short amount of time on a page, it was considered that the student could not have read this page in such a short amount of time. This gave evidence that the student had visited such pages in her/his effort to find an appropriate page to study and hence s/he did not spend much time on these pages that were not appropriate. In contrast, when a student had spent longer periods of time when reading a page, it was considered that this page interested her/him and was appropriate for her/his level. Thus, in the experiment, the more pages a student had visited for very short time on each of them, the more disoriented s/he was considered. This kind of evidence was used to evaluate the navigational support the systems gave to students.
To achieve a comparison of the two systems in the navigational support we compared the percentage of the theory pages visited by the students of the two groups for less than half a minute. The amount of time of half a minute was considered as the shortest amount of time needed by a student to read a page. Thus, any amount of time less than half a minute was considered as not sufficient for a student to read the content of a page. The decision about the amount of time that would be the threshold (in this case, half a minute) was made with the collaboration of human teachers. The pages that the student visited for less than this threshold were considered not interesting, not ready or already known to the student. According to the results of the study the students of the ICALL group visited such pages in a percentage of 18.02%, whereas the students of the CALL group in a percentage of 36.31%. This result is overly significant (t(100)=-7.45, p<0.0001) and allowed us to draw the conclusion that the students of the ICALL group were assisted by the adaptive navigation support in order to select theory pages that were suitable for their knowledge level. However, the percentage of not suitable pages was not as low as we expected. This might have been due to the fact that students did not interact with the system for a very long period of time and their exploration actions to learn how the system works were counted as actual learning behavior.

Finally, the system’s adaptation decisions that aim at helping students when solving exercises were evaluated based on the students’ requests to see the solution to exercises and the number of correct student answers that were given after the system’s advice to students’ errors. As we can see in Table 1, the students of the ICALL group outperformed the students of the CALL group in correcting their erroneous answers based on the system’s feedback (45.39% versus 26.46%). This result was also statistically significant (t(100)=6.53, p<0.0001). Similarly, the comparison between the number of students’ requests to see the solution to an exercise that they were asked to solve revealed that the students of the ICALL made significantly fewer requests than the students of the CALL group (t(100)=-6.62, p<0.0001). These results showed a trend in favor of the ability of Web-PVT to perform error diagnosis and provide individualized feedback to errors.

4.2. Usability evaluation results

Web-PVT was evaluated with respect to its usability and friendliness, based on the answers of the students of the two groups to the questionnaire that was given to them in their last session. The questions in the questionnaire mainly concerned usability and friendliness issues of the systems. The answers of students to these questions were meant to reveal how the students liked their interaction with the respective version of the system. In particular, the students’ scores to the scaled-answer questions were used in order to calculate the mean scores of each group. Then, the mean scores were analyzed and compared using again t-test statistics, with the alpha value set to 0.05. The questions and the results of this analysis are summarized in Table 2.
Table 2. Results of the usability and friendliness evaluation.

<table>
<thead>
<tr>
<th>Question</th>
<th>ICALL Group (n=51)</th>
<th>CALL Group (n=51)</th>
<th>t(df); p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Value</td>
<td>Standard Deviation</td>
<td>Mean Value</td>
</tr>
<tr>
<td>How much time did you need to get familiar to using the system? (ranging from 1: too little to 5: too much)</td>
<td>2.69</td>
<td>0.65</td>
<td>1.90</td>
</tr>
<tr>
<td>Did you find the system easy to use? (ranging from 1: very difficult to 5: very easy)</td>
<td>3.57</td>
<td>0.76</td>
<td>4.12</td>
</tr>
<tr>
<td>Did you enjoy studying with the system? (ranging from 1: not at all to 5: very much)</td>
<td>4.12</td>
<td>0.71</td>
<td>4.24</td>
</tr>
<tr>
<td>Was it easy to study theory using the electronic textbook? (ranging from 1: very difficult to 5: very easy)</td>
<td>3.31</td>
<td>0.79</td>
<td>1.55</td>
</tr>
<tr>
<td>Are you satisfied from the system’s feedback to your mistakes? (ranging from 1: not at all to 5: very much)</td>
<td>4.10</td>
<td>0.67</td>
<td>1.28</td>
</tr>
<tr>
<td>An hour of tutoring with the system is more constructive than an hour of conventional tutoring in a school classroom. (ranging from 1: totally disagree to 5: totally agree)</td>
<td>1.33</td>
<td>0.52</td>
<td>1.37</td>
</tr>
</tbody>
</table>
To a large extent, the results were in accordance with the findings of other similar studies. A first conclusion that could be drawn from the results presented in the table is that the students of the ICALL group needed more time to get familiar to using the system (mean score in the first question: 2.69) than the students of the CALL group (mean score in the first question: 1.90) and that they found the system harder to use. This particular result was expected due to the fact that the adaptive annotation technique used for helping the student in her/his navigation adds more features to the interface that the users have to get acquainted with. These additional features could lead to additional cognitive burden for users who could be distracted to some extent from the content. In fact, this is a risk for every system that makes use of the adaptive annotation technique. However, in our study, the differences in the students scores in the first two questions were not statistically significant. This means that the additional cognitive load to students of the ICALL group was not very important in comparison with the non-intelligent system that did not have any extra features in its user interface. Furthermore, both groups rated their enjoyment of the system on a similar scale. In particular, both systems achieved high scores (4.12 the ICALL version and 4.24 the CALL version). This result can be explained based on the students’ enthusiasm about the change in the regular way they were being taught English.

However, the students of the ICALL group were significantly more satisfied from the system’s ability to provide individualized support when they studied theory and solved exercises (third and fourth question of the table). Indeed, the individualized feedback to errors provided by the intelligent version of the system seemed to have helped the students who rated it in a highly positive way (mean score 4.10). These results are in accordance with the findings of the analysis of the objective data gathered in the personal student log-files (section 4). Finally, the students of both groups did not consider the tutoring with the system similarly constructive as the conventional tutoring in a school classroom with a human tutor. This result is not surprising due to the fact that Web-PVT was not designed to be used alone but rather constitute a tool to aid human teachers and students in the tutoring of the domain of the passive voice of the English language.

5. Discussion and Conclusions

The primary aim of educational software is to help students learn. Therefore, evaluation of this kind of software using real students is crucial. In this paper we present the results of an evaluation study of Web-PVT, which is a Web-based ICALL system. In particular, we aimed at evaluating Web-PVT along three dimensions: a) the effect of the intelligent features of the system on the learning outcome of students, b) the system’s ability to provide individualized support to students that leads to more effective use of the system and c) the general usability and friendliness of the ICALL. For these reasons, we compared the system with a second version of it that did not provide intelligent instruction.

The results of the empirical evaluation of Web-PVT were very encouraging and showed that intelligence helped students increase their knowledge of the domain. Indeed, the students who used the version of the tutor that incorporated intelligence performed significantly better in the final test provided after their interactions with the system as compared with the students who used the version that was not intelligent. This particular finding is important since many prior studies did not show the same results.
A second important advantage of the intelligent version of the tutor was that it allowed student to interact with the system in a more efficient way. More specifically, the system’s ability to provide adaptive navigation support to students when they studied theory seemed to have a positive effect in making their paths less linear and more explorative. This result was not surprising, due to the fact that it is one of the most common finding of most empirical studies that evaluate such kind of adaptive systems. Furthermore, the individualized feedback to errors provided by the intelligent version of the system helped students correct their mistakes and was rated in a very positive way.

However, the students who used the intelligent version of the system found it harder to use and needed more time to get familiar to using the system. This finding, although negative was expected since the intelligent features add a component on the user interface and in many cases may result in a greater cognitive overload. Furthermore, due to the limited time that students had for their interaction with the system, some of the results were biased by explorative student actions. Indeed, in the future we plan to investigate the effects of intelligence in longer terms. Furthermore, more studies should be conducted in order to generalize the results of the evaluation reported in this paper. In the future evaluations, more students and teachers should participate in the experiments. These students should be older than the students who participated in this experiment, so that we could investigate whether similar results stand for students of different ages.

References


