Abstract—In this paper we describe the method for initializing the student model in a Web-based language tutor. This tutor is an Intelligent Tutoring System (ITS) that operates on the Web and aims at teaching non-native speakers the domain of the passive voice of the English language. It uses an innovative combination of stereotypes and the distance weighted k-nearest neighbor algorithm to initialize the model of a new student. In particular, the student is first assigned to a stereotype category concerning her/his knowledge level based on her/his performance on a preliminary test. The system then initializes all aspects of the student model using the distance weighted k-nearest neighbor algorithm among the students that belong to the same stereotype category with the new student. The basic idea of the algorithm is to weigh the contribution of each of the neighbor students according to their distance from the new student; the distance between students is calculated based on a similarity measure. In our case the similarity measure is estimated taking into account the students’ mother tongue, how careful they are when solving exercises, as well as their knowledge of other languages. This information is acquired directly by the student at her/his first interaction with the system.

Index Terms—Intelligent Tutoring Systems, Student Model Initialization, Distance Weighted k-Nearest Neighbor, Stereotypes.

I. INTRODUCTION

INTELLIGENCE and adaptivity in educational computing is provided by Intelligent Tutoring Systems (ITS). To a large extent in ITSs, intelligence and adaptivity are achieved by the incorporation of a student modeling component. The ITS should have some information about the individual characteristics of the student. Such information may be personal, domain independent data, such as the student's name, age, mother tongue, etc., as well as several domain-related student attributes, for example the student's knowledge level of the domain being taught.

When students start working with an ITS, the system has no prior knowledge about their proficiency level of the domain nor of their learning characteristics. However, the student model attempts to provide individualized support. Therefore, the initialization of a student model is of great importance because misleading messages that do not correspond to the real strengths and weaknesses of the student may cause them frustration. Indeed, an ITS runs the risk to lose its credibility and be considered as irritating and worthless to use by a student, if it fails to make plausible hypotheses about a student, before the student loses her/his patience with the system.

An obvious strategy for acquiring initial information about a student’s knowledge level and misconceptions is to pose appropriate questions to the student in an initial phase of the system’s usage. This approach may be applicable in cases where the domain of interest is rather restricted. However, in most cases, such an approach would expect the student to answer to questions for a long period of time before s/he could actually start working with the system. In order to overcome this drawback, the problem of initializing the user model is often solved by assigning the user to a certain group of users, as is the case in the stereotypes approach (e.g. [1], [2], [3], [4]). Stereotypes constitute a very powerful technique in providing information based on few observations. However, they do not permit the formation of a student model that focuses on special individual characteristics, e.g. the mother tongue of the student. Furthermore, stereotypes are constructed by hand before real users have interacted with the system and are not updated until a human does so explicitly.

In this paper we describe the approach taken by Web Passive Voice Tutor (Web PVT) to initialize the student model. Web PVT is an adaptive and intelligent web-based tutoring system that aims at teaching non-native speakers the domain of the passive voice of the English language. Our approach to student model initialization exploits the fact that Web-based systems have a large number of users and we use a machine learning reasoning mechanism that is based on recognized similarities between users. The initialization of the student model is performed dynamically for each student after they have interacted with the system and are not updated until a human does so explicitly.

According to Sison and Shimura [5], machine learning or machine learning-like techniques have so far been used in two areas of student modeling research. First, many systems, such as DEBUDDY [6], ACM [7], THEMIS [8] and ASSERT [9], have used methods inherited from the area of machine learning in order to induct a single, consistent
student model from multiple observed student behaviors. Furthermore, machine learning or machine learning-like techniques have also been utilized for the purpose of automatically extending or constructing from the bug library of student modelers. Systems that have followed such an approach include PIXIE [10], Hoppe's system [11] and MEDD [12], [13].

In the case of Web PVT we have used machine learning in order to address the problem of initializing the student model taking into account the issue of language transfer. In particular, we use an innovative combination of stereotypes [1], [14] and the distance weighted k-nearest neighbor algorithm [15], [16], [17] to set initial values to all aspects of the student model. 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 refine the estimation about the student's knowledge level of each concept and her/his proneness to make mistakes concerning this concept. 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 enough for the system to have been able to construct their individual student models inferred from extensive observations of their behavior. The basic idea of the algorithm is to weigh the contribution of each of the neighbor students according to her/his distance from the new student; the distance between students is calculated based on a similarity measure. In Web PVT the similarity measure is estimated taking into account the students’ mother tongue as well as their prior knowledge of other languages.

II. OVERVIEW OF WEB PVT

Adaptivity is an essential matter in web based educational software, since distance learning systems are aimed to be used by many different students and in situations where no teacher is available to help users in their learning process. The student model is mainly used to provide adaptivity to the system. In this section we briefly discuss the adaptation decisions that are made by Web PVT based on the model of each individual student [18]. Moreover, we give a brief description of the domain knowledge of Web PVT.

A. Functionality of Web PVT

Web PVT incorporates techniques from Intelligent Tutoring Systems and Adaptive Hypermedia to tailor instruction and feedback to each individual student. In particular, based on the information contained in the student model the system supports the student while 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.

Moreover, in order to further individualize instruction, Web PVT consults the student model before presenting the student with new exercises to solve. In order to select the next exercise to present to the student, Web PVT consults the individual long term student model. In case there is one or more concepts that are marked as "read" but are not "known" to the student, the tutor selects an exercise that evaluates as many of those concepts as possible. A concept is considered "not known" either due to the fact that the student has never before solved exercises associated with this concept, or because the percentage of correctly solved exercises associated with the concept is unsatisfactory. If there is no such concept that the student has read and still faces difficulty in using while solving exercises, the system selects a concept for which the student has the lowest percentage of correctly answered exercises and presents an exercise testing this concept.

Finally, the student model is also utilized in order to perform error diagnosis and advice generation in cases of error when solving exercises. When the system performs error diagnosis, it takes into account information that has been collected about the specific student who has made the mistake. 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. Furthermore, the student's level of knowledge is used in order to determine the amount of help to be provided to the student while s/he is solving exercises. The less proficient a student is the more help s/he gets.

B. Domain Knowledge Representation

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 while formulating the student model.

Similarly with KNOME [19] the grammatical concepts are grouped in categories based on their level of difficulty. These stereotype categories include simple, mundane and complex concepts. In Web PVT the identification of the stereotype hierarchy was a result of the empirical study conducted before the system's development.

Each node in the domain knowledge represents a certain category of concepts, 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 tense" concept. A precondition relation points from a concept to another, which is its prerequisite. For example, in order to master the simple past, one should know how to form irregular verbs.
III. REPRESENTATION OF THE STUDENT MODEL

In order to construct a student model, information about the student should be acquired. Such information may be personal, domain independent data, such as the student's name, age, mother tongue, etc., as well as several domain-related student attributes, for example the student's knowledge level of the domain being taught. For some student data (e.g. the mother tongue of the student and other languages s/he may already know) direct provision by the student is the only possible source of information. Such information is acquired by Web PVT using a set of questions posed to the student in the initial phase of the system's usage. Furthermore, Web PVT also uses a preliminary test in order to assess the initial knowledge level of the student concerning the passive voice of the English language. The test has been constructed by human experts so as to contain representative questions that cover the whole domain of the passive voice of the English language. The preliminary test is given to students before they have ever interacted with Web PVT.

When the necessary information about the student has been acquired, the system needs to properly represent the student characteristics so that they could be further exploited. In the case of Web PVT, the student model is represented as a set of feature vectors. The first vector is responsible for representing information acquired by the student in her/his first interaction with the system. The characteristics contained in the first vector include the name of the student, her/his age, the stereotype category that s/he belongs, 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. Therefore, the first vector is defined as follows:

\[
\text{<Student\_Code, Name, Age, Stereotype, Carefulness, Mother\_Tongue, Language1, Language2, …>}\]

The second vector is directly related to the domain knowledge of Web PVT. In particular, for each concept in the domain knowledge 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 values of the estimations are within the range \([0..1]\). When estimating degree of knowledge of a concept, 0 depicts the system's belief that the student does not know at all the concept, while 1 represents the system's belief that the student knows this concept very well. Furthermore, when referring to error proneness concerning a concept, 0 represents the system's belief that the student never makes mistakes when using a concept, and 1 represents the system's belief that the student always makes mistakes related to the concept. Therefore, the second vector is defined as follows:

\[
\text{<Student\_Code, Know\_Concept1, Errors\_Concept1, Know\_Concept2, Errors\_Concept2, …>}\]

If the domain of the passive voice of the English language consisted only of three concepts, namely "Verb Tenses", "Agentless Passive", and "Agent Connecting Words", the pair of feature vectors:

\[
\text{<Stu\_1, Jim Kotronakis, 14, Intermediate, Careless, Greek, German, Russian>, and}\]

\[
\text{< Stu\_1, 1, 0.1, 0.7, 0.3, 0.8, 0.1>}\]

would represent a student named Jim Kotronakis who is 14 years old. Jim is considered by the system as an "intermediate" in the domain of the passive voice and is careless while solving exercises. Furthermore, the student's mother tongue is Greek and he also knows German and Russian as foreign languages. The second vector depicts the system's beliefs about the student's knowledge. In particular, the system believes that Jim's degree of knowledge of the concept "Verb Tenses" is 1 and that he makes mistakes while using this concept in a percentage of 10%. In addition, Jim's degree of knowledge of the concept "Agentless Passive" is 0.7, whereas the student's proneness to make mistakes while using this concept is 30%. Finally, the system estimates that Jim knows the "Agent Connecting Words" concept in a degree of 0.8 and he makes mistakes in this concept in a 10% percentage.

IV. ARCHITECTURE OF THE INITIALIZATION MODULE

In this section we describe the architecture of the module of Web PVT that is responsible for initializing the model of a new student (Fig. 1). First, personal characteristics and the knowledge level stereotype category of the student are acquired based on an interview and a preliminary test posed to the student when s/he registers to the system. This information is represented as a feature vector (described in the previous section). Then, based on the models of other students that belong to the same stereotype category with the new student, the system produces a second vector that represents the system's estimations of the degree of knowledge and error proneness of the student for each concept in the domain knowledge. In our approach to produce
the second domain related feature vector, we take weighted sums of known values to produce a value for an unknown quantity. In particular, for each unknown estimation of the degree of knowledge or error proneness of a concept, the known values are the estimations of the degree of knowledge or error proneness of other students that belong to the same stereotype category with the student in question. The weights are a measure of the similarity between the student in question and the other students of the stereotype category. In case there are no students belonging to the same stereotype with the new student, then her/his student model is initialized using the default assumptions of the active stereotype.

V. STEREOTYPES

The main components of a stereotype are [20]:

1. a body, which contains information that is typically true of users to whom the stereotype applies, and
2. a set of activation conditions (triggers) for applying the stereotype to a user.

In Web PVT, the student is classified into stereotypes concerning the knowledge level of the student in the domain being taught. In particular, there are four distinct stereotypes, namely novice, beginner, intermediate and advanced. Similarly with ELM-ART II (Weber and Specht 1997), students are initially assigned to one of the four stereotypes depending on their performance on the preliminary test posed on the student's first interaction with Web PVT.

The trigger conditions that are sought in the preliminary test depend on the students' errors. For example in the case of a beginner student, a trigger condition may occur when s/he commits errors in the verb tense conversion to the passive voice in a percentage that is greater than 60%. That is because the concept of the verb transformation is considered mundane according to the domain knowledge hierarchy. As the system acquires more information about a particular student, it may use it to alter the active stereotype for that student. To do this, the system uses the retraction conditions of the stereotypes. In the case of the beginner stereotype, a retraction condition may occur when a student makes mistakes in irregular verbs in a percentage greater than 30%. In this case, the beginner stereotype would be deactivated and the novice stereotype would be activated instead. This is so because the irregular verbs are considered a precondition concept to the domain of the passive voice.

In Web PVT, stereotype inferences concern default assumptions about the student's mastery of certain grammatical concepts. In particular, for each stereotype category concerning the student's knowledge level there are default assumptions that are associated with each one of the categories of the grammatical concepts (namely simple, mundane and complex) of the domain. These assumptions have been based on an empirical study that involved human teachers and students. For example, the default assumptions of the intermediate stereotype would be that the student already has satisfactory knowledge of all the concepts that are considered simple, s/he has some knowledge of the mundane concepts but seems to face difficulty in using those concepts while solving exercises, whereas the student does not know the concepts that are considered complex.

VI. DISTANCE WEIGHTED k-NEAREST NEIGHBOR ALGORITHM

The distance weighted k-nearest neighbor [15], [16], [17] is a refinement of the original k-nearest neighbor algorithm [21], [22]. In general, nearest neighbor learning algorithms typically store all of the n available training examples during learning. These algorithms use a distance function to determine how close a new query instance is to each stored instance, and use the nearest instance or instances to classify the query instance [23]. The basic idea of the distance weighted k-nearest neighbor algorithm is to weigh the contribution of each of the k neighbors according to their distance to the query point, giving greater weight to closer neighbors[24].

The main decisions that have to be made while applying a distance weighted nearest neighbor algorithm are the following:

1. The features that would be used to formulate the input space of the distance function have to be selected.
2. A distance function must be chosen to estimate the similarity between two instances.
3. The number of neighbors (k) that would participate in the classification task should be defined.
4. A function has to be designed in order to classify new instances.

In the case of Web PVT the distance weighted k-nearest neighbor algorithm is used in order to initialize the model of a new student taking into account the issue of language transfer and how careful the student is while solving exercises. The choice of the attributes that are used in order to measure the distance between students was based on the fact that students often use their mother-tongue (or possibly some other foreign language distinct from the target language) experience as a means of organising the second language [25]. Furthermore, the degree of carefulness of a student may be a way to explain certain category of mistakes. Therefore the student characteristics that are used in order to measure the distance between two students are the stereotype category of the student, the estimation of how careful a student is while solving exercises, her/his mother tongue, as well as the foreign languages the student already knows. The input vector of the distance function would be of the form:

< Stereotype, Carefulness, Mother_Tongue, Language1, Language2, …>

Due to the fact that the student characteristics that are chosen to formulate the input space have all nominal values, there is a need to use a distance function that can handle such values. In the case of Web PVT we use a simple overlap metric similarly with the distance metric for nominal values used in IB1, IB2, and IB3, which are systems described in [26]. Unknown attribute values are handled by returning an attribute distance of 1 (maximum distance) in case either of
the attribute values is unknown. Therefore the distance function used by Web PVT to calculate the distance between two values \( x \) and \( y \) of a given attribute \( a \) is defined in (1).

\[
d_a(x, y) = \begin{cases} 
1, & \text{if } x \text{ or } y \text{ is unknown}, \\
\text{overlap}(x, y) & \text{else}
\end{cases}
\]  

(1)

Furthermore, the \( \text{overlap} \) function is defined as shown in (2).

\[
\text{overlap}(x, y) = \begin{cases} 
0, & \text{if } x = y \\
1, & \text{otherwise}
\end{cases}
\]  

(2)

The overall difference measure of two students \( s_x \) and \( s_y \) is then calculated as defined in (3), where \( n \) is the number of attributes that are used to measure the distance between two students.

\[
\Delta(s_x, s_y) = \sum_{a=1}^{n} d_a(x_a, y_a)
\]  

(3)

Another important decision that should be made is the definition of the number of neighbors that would participate in the classification task. In our approach, we have set the number of \( k \) to be the number of students that belong to the same stereotype category with the student in question. This is due to the fact that students that belong to different stereotypes are not expected to have similar knowledge of the domain, irrespective of their mother tongue or the foreign languages they may know. For example, intermediate and advanced students may have the same knowledge of some simple concepts. However, they are not expected to have a similar degree of knowledge in concepts that are considered complex.

Finally, a classification function has to be chosen that uses as input the instance that should be classified (new student) and the \( k \) nearest neighbors (students that belong to the same stereotype). In the case of Web PVT, the system sets initial values to the estimations of the student's degree of knowledge and error proneness for each concept in the domain based on the known values of these attributes that are acquired by students that have already been registered in Web PVT. In order to predict the values of the degree of knowledge and error proneness of the new student \( (s_q) \), we use a distance weighted mean value of the degree of knowledge and error proneness of the \( k \) students that belong to the same stereotype with the new student \( (s_1, s_2, ..., s_k) \). The vote of each of the neighboring students is weighted according to the inverse square of its distance from \( s_q \). Therefore, for each concept in the domain, the function that estimates the degree of knowledge of the new student \( (s_q) \) is presented in (4).

\[
\text{Know(Concept}_x, s_q) \leftarrow \frac{\sum_{i=1}^{k} w_i \text{Know(Concept}_x, s_i)}{\sum_{i=1}^{k} w_i}
\]  

(4)

where \( w_i \) is the weight of the contribution of each student and calculated using (5).

\[
w_i \equiv \frac{1}{\Delta(s_q, s_i)^2}
\]  

(5)

To accommodate the case where the query student \( s_q \) exactly matches one of the students \( (s_i) \) that are used as training examples and the denominator \( \Delta(s_q, s_i) \) is therefore zero, we assign \( w_i \) to be equal to 1 (maximum weight) in this case. The error proneness of the new student concerning a concept \( \text{Error_proneness(Concept}_x, s_q) \) is estimated in a completely similar way.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we have described the approach taken by Web PVT to initialize the model of a new student based on recognized similarities with other students that have already interacted with the system. In Web PVT the similarity is estimated taking into account the students’ mother tongue, their degree of carefulness when solving exercises, as well as their prior knowledge of other languages. In this way, the system constructs an initial student model that addresses the important issue of language transfer. The importance of this issue stems from the fact that students often use already acquired knowledge while learning a new subject. Our approach to student model initialization uses an innovative combination of stereotypes and the distance weighted k-nearest neighbor algorithm. Stereotypes are used to make initial hypotheses about the knowledge level of the student, whereas the distance weighted k-nearest neighbor algorithm is utilized in order to refine the estimations of the student's knowledge level of each concept and her/his proneness to make mistakes concerning this concept, based on the student’s similarity with other students of the same stereotype category.

Within the future plans of this research is the evaluation of the approach using real students in order to prove its usefulness in terms of educational effectiveness. This kind of evaluation needs a lot of time and a lot of students, who will use the system at different time periods. In this way, we will be able to evaluate the student modeller’s ability to learn from the students that have already interacted with system several times. Furthermore, the initialization approach described in this paper is quite domain independent and could be used to form a general framework for initializing student models. Indeed, we plan to apply this initialization approach to a very different tutoring domain (Mathematics) in order to test its generality.
REFERENCES


