Development of a System for Computer-assisted Learning of Croatian Orthography Concerning the Yat Reflex (CAL-COR)

Dolores Grmača
University of Zagreb,
Department of Information Science, Faculty of Philosophy
dgrmaca@ffzg.hr

Ozren Kubelka
Median et d.o.o., Zagreb
ozren@kubelka.net

Nikola Ljubešić
University of Zagreb,
Department of Information Science, Faculty of Philosophy
nljubesi@ffzg.hr

Abstract: *This paper describes the development of the system for computer-assisted learning of Croatian orthography concerning the yat reflex (CAL-COR). Our previous research resulted in an online example-based system (http://calcor.ffzg.hr) with already two versions that are described in this paper. The improvement of the system includes changes in the algorithm that takes control of the learning process in a greater fashion. Further improvement is made in the construction of the user's profile which will enable further research and a more specific approach to the learning process concerning user's age, foreknowledge and learning abilities. Also, a visibility plan is developed aiming to popularize such system in educational institutions and broader. Frequent usage of the system is of great importance for collecting data which will enable further optimization of the system.*

Keywords: *computer-assisted language learning (CALL), web application, Croatian orthography, reflex yat*

1. INTRODUCTION

As Internet applications become more and more advanced, using Web-based technology to enhance language learning has become a very popular practice in the world [8] [14]. One of the most compelling areas of exploration for computer use is the field of language learning [11]. Learning language through technology has become an important part not only in applied linguistics [5], but also in computer and information science, cognitive science, psychology and many other disciplines what makes this field extremely cross-disciplinary. That is why CALL researchers and developers find themselves at the crossroads among all these disciplines [3]. The European Commission makes great efforts to create a "virtual European education area" involving cross-cultural cooperation [4]. Our project tries to contribute to these efforts.

This paper presents a Web-based system for learning Croatian orthography concerning the *yat* reflex which is the product of our previous research [1]. One of the biggest problems in the Croatian orthography is the *yat* reflex. Otherwise, the Croatian orthography is simple due to the high degree of the phonetization of the script. In the past, the Croatian language had the vowel *yat* (ě) that could be both long and short (like today's vowels). In contemporary Croatian language there are four possibilities for its rendering (*yat* reflexes) – *ije*, *je*, *e*, *i*. The *yat* reflex gets replaced due to the change in the quantity of the syllable which includes it. That is the reason why the *yat* reflex is considered such a complex problem. This claim is also proved by the fact that students of senior years at the Faculty of Philosophy (many of them students of Croatian language and other philological studies) made mistakes in 16,66% of cases during our previous research [1].

Most papers in the field of Computer-assisted/aided language learning (CALL) describe such systems as an additional tool in a classroom education environment [5] [13] [12]. We believe that our system can be used not only in that fashion, but also as a tool everyone can use on his own, regardless the age and prior education.

The purpose of this paper is to describe the development of a system that should move towards an intelligent system for language learning. Such system should be able to personalize the learning process in greatest fashion which is the modern striving of all intelligent systems for Web-based education [2] [9]. Our system is already capable of diagnosing user's problems in the field of the *yat* reflex and leading the learning process in that direction which is also one of the crucial characteristics of intelligent computer-assisted learning (ICALL) [10].

In the next two sections we will describe two previous versions of the system and the third one that tries to eliminate some imperfections we experienced in the previous two. We will also provide a visibility plan whose aim is to popularize our system in educational institutions and wider. That should provide us with data we need to evaluate our system and continue its development.

2. FIRST TWO VERSIONS OF THE SYSTEM

The system was developed as an ASP.NET web application [7] [15]. The first version, CALCOR 0.1 used a Microsoft Access database, while the second one, CALCOR 0.2, as well as the third one, uses a MS SQL Server 8.0 database [6].

The Access database consisted of five tables. In the first table data about the users was stored. In the second one the users' answers were stored. The main aim of this table was collecting statistical data about the usage of the system which would enable its evaluation and further improvement. In the third table data about users' mistakes was stored. This data navigated the future learning process of every user. In the fourth table 85 rules concerning the *yat* reflex were listed. In the fourth table 2005 word forms were stored. Every word form had a foreign key that pointed to its respective rule. There were 393 word forms that didn't have any rule attached to them because there is no rule which can explain them. The word forms were also grouped into classes of words with a similar or same stem. There were 228 such classes.

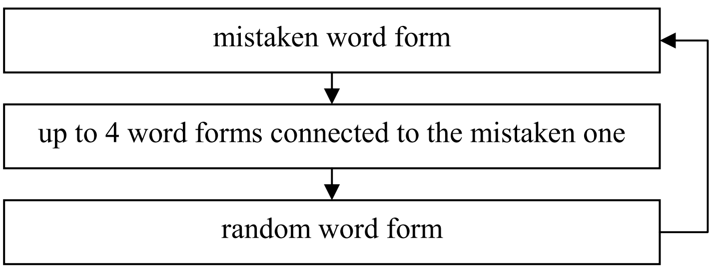
The database structure didn't change in the second version of the system, it just moved from a Microsoft Access database to the more sophisticated MS SQL Server 8.0 database. The structure of the tables is shown in table 1. The algorithm of the system did experience some changes in moving from version one to version two. In both versions the main approach in picking a word form is the random method. The user gives answers to the questions by filling in the gaps selecting the correct *yat* reflex from a combo box.

In the first version CALCOR 0.1 a word form randomly picked from the table [t_mistakes] is given to the user. The algorithm uses the Random .NET class for choosing a random record from the database. Every time the user answers a question, he is notified whether his answer is correct or not. If the answer is not correct, also the correct word form is given. After the first word form, up to four randomly picked word forms are given to the user. Those word forms are connected with the first one via the same rule or the same or similar stem. At the end of this circular algorithm, one completely random word form is given to the user. The aim of this word form is to ensure that all word forms are included in the learning process, not only the mistaken ones and the ones connected to them. After that word form, the circular algorithm starts from the beginning picking randomly a word form from the list of the user's mistakes. The list of the mistakes is filled for the first time for a specific user at the moment a new account is created with 100 especially complex word forms concerning the *yat* reflex. Each time the user makes a mistake, the mistaken word form is added twice to his list of mistakes. Each time the user answers correctly to a word form that is in his list of mistakes, one record is being deleted from that list. The learning process is considered to be over when there are no more records in the table [t_mistakes] for the specific user. A graphical representation of this cyclic algorithm is shown in picture 1.

Attribute name	Data type	Description
t_user		
id_user	int (identity)	ID of the user
user	nvarchar	Username
passwd	nvarchar	Password
option	tinyint	When is the respective rule shown
t_answers		
id_answer	int (identity)	ID of the answer
answer	nvarchar	Answer to the question
correctness	tinyint	Is the answer correct
id_form	int	ID of the question answered
id_user	int	ID of the user answering the question
t_mistakes		
id_mistake	int (identity)	ID of the mistake
id_user	int	ID of the user
id_form	int	ID of the question
t_rules		
id_rule	int (identity)	ID of the rule
rule	nvarchar	Rule
example	nvarchar	Example of the rule
t_forms		
id_form	int (identity)	ID of the word form
form	nvarchar	Word form
id_rule	int	ID of the respective rule
stem	int	Class of word forms with same or similar stems

Table 1 – Structure of the tables in the CALCOR 0.2 system

The algorithm in the second version of the system CALCOR 0.2 differs from CALCOR 0.1 on a technical level in the way the random records from the database are picked. While in the first version of the system the Random .NET class was used, the second one uses a specific SQL Server command for it. The SQL command "SELECT TOP 2 id_form FROM t_mistakes WHERE id_user = 3 ORDER BY NewID();" selects two random word forms from the table [t_mistakes] made by the user which ID is 3. This method of picking random records gives much better results than the one used in the first version of the system.



Picture 1. Graphical representation of the cyclic algorithm in CALCOR 0.1

In the second version of the system, CALCOR 0.2, when the user makes a mistake, two records are written in the table [t_mistakes], but the user is prompted for the correct *yat* reflex of the same word form once again. If the user answers correctly, which is more than likely because he got the correct word form right after making the mistake, one of these two records is being deleted right away.

In the next step, only up to two random word forms connected to the mistaken one are given to the user. In our opinion the number of up to four word forms in CALCOR 0.1 was too high. These word forms are mostly not the word forms the user has mistaken, so they should not be of central importance for us.

After that step, another completely random word form is picked with the same aim as in the first version of the system – to guarantee that all word forms from the database are part of the learning process. At this point, the circular algorithm starts from the beginning picking up a random word form from the table [t_mistakes].

Another difference between the first and the second version of the system is that in the second version the user is being informed all the time on the specific algorithm stage. We experienced the lack of this information in the first version of the system as very confusing because the user did not know why he is given the particular word forms. We find that this innovation improves the transparency of the system.

Both versions of the system did not have any intelligent way of ending a learning session, the user was prompted after he answered fifty questions if he wants to stop learning and come back another time or if he will continue learning. There was little of statistical data shown to the user – his average accuracy and the number of his answers in total as well as his average accuracy and the number of his answers during the specific session. This data can be easily used to follow the user's success in answering the given questions during a learning session.

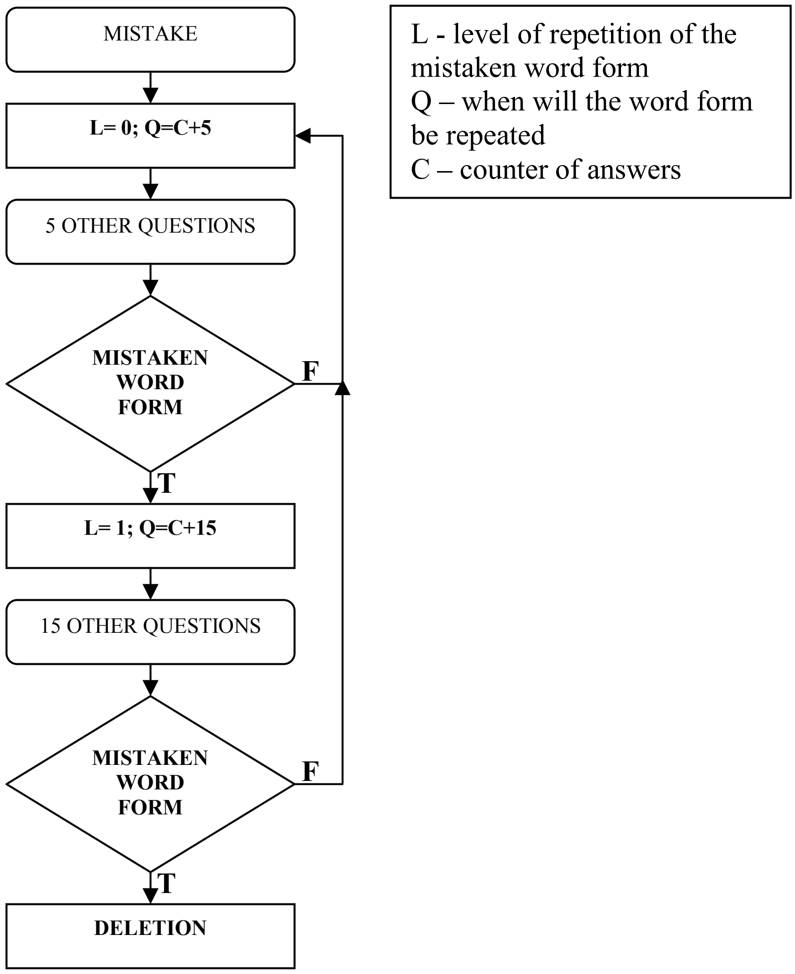
In our database every word form is connected to a specific rule. That enables the system to show the specific rule while asking the user a word form. The user can choose between three options anytime – the rule is shown after answering the question, the rule is shown before and after answering the question and the rule is not shown at all. The decision about how much the user will be exposed to the rule is left to the user only. Namely, on one hand our previous research showed that the deductive, rule-based approach in some cases produced least results [1]. On the other hand, we strongly believe that rules are an important part of learning orthography and language in general.

We felt that the weakest link of the whole system is the algorithm that relies on the random algorithm in picking the mistaken word forms. The algorithm should be more sensitive to the time span between a mistake and the repetition of the mistaken word form. There is also no data collected about the user's profile such as age and previous education. Furthermore, the system is not sensible enough on changes in user's success during one learning session. These changes should, among other things, indicate when one learning session should come to its end. These are our biggest improvements made in the third version of the system described in the following section.

3. THE THIRD VERSION OF THE SYSTEM – CALCOR 0.3

The biggest improvements made in the third version of the system in comparison to the first two are the refinement of the algorithm in means of greater control over the time span between a mistake and the repetition of the mistaken word form, an expansion of the user's profile and greater sensibility to the changes in user's success during one learning session.

In the third version of the system CALCOR 0.3 when the user makes a mistake, not only the user's ID and the ID of the specific word form are written in the table [t_mistakes], but also the value of a counter of complete number of answers – "id_question". This value ensures that the algorithm knows how many answers have passed between the last repetition of a mistaken word form. Furthermore, a value indicating the level of repetition of the mistaken word form "id_level" is written in the same record. The main idea of the new algorithm is to repeat the mistaken word forms until they are not answered correctly twice in a row in a span of 5, e.g. 15 answers. When the user makes a mistake, this word form is put into the table [t_mistakes] with the value of the counter of answers increased by 5. That value indicates when this word form has to show up again as a question. The value "id_level" is put to 0. When the specific word form appears as a question again, if it is answered correctly, the value of "id_question" is put to the present value of the counter of answers increased by 15. The value of "id_level" word form is put to 1. If the user answers the question correctly after fifteen more other word forms, this record is deleted from the table [t_mistakes]. If the user answers incorrectly one of these two times, the data in the specific record is put to its initial values – the present value of the counter increased by 5 in "id_question" and 0 as the value of "id_level". A graphical representation of this algorithm is shown in picture 2.



Picture 2 – Graphical representation of the algorithm for mistakes handling in CALCOR 0.3

The success of the user in answering the questions during one learning session is monitored all the time. After every 12 questions, an evaluation screen is showed. On these evaluation screens the user is informed about his success to that point and encouraged to do better or congratulated depending on his results. The aim of these evaluation screens is to live up the learning process and to motivate the user to use all his potentials. If the user starts answering in a bad fashion right at the start and does not improve in the next twelve answers, he is asked if he wants to leave the system and come back another time when he is readier to learn. If the user is answering with average or better correctness for some time, but then starts answering more and more incorrect and does not improve in the next twelve answers, he is asked the same question. The user is not forced to leave the system, this decision is left to him only.

In this version more data is collected from the user while registrating on the system. In CALCOR 0.1 and CALCOR 0.2 only a username and a password were required. In CALCOR 0.3 users have to register themselves through a registration form. The registration requires information about user's year of birth, level of education, sex, e-mail address and an optional comment. We believe that a longer list of required data would irritate potential users and make some of them not register. Such user data will expand the functionality of the system in two ways: it will enable our further research that should result in a system taking different approaches depending on user's age and education, but this data will also enable research on a wider perspective. We will be able to draw conclusions that will be valuable for the field of CALL, e.g. ICALL, but also education and language learning in general. It will be possible to systemize and select data through many criteria.

To be able to collect data needed for this research the system has to be used by many users. The system was used at this point by about 100 users who answered about 8000 questions. To enlarge the number of users and answers, we believe our project needs a visibility plan that should make our system visible to the general public. Our plan is to contact institutions such as the *Ministry of Science, Education and Sports*, faculties, primary and high schools using the top-down approach. We will also contact individuals such as teachers of Croatian language, university professors, students, pupils etc. using the bottom-up approach. We will contact these institutions and individuals through official channels. For potential users who are not informed through official channels we are planning conventional methods of promotion such as articles in the media and on the Internet, paper ads, internet banners etc. Namely, the aim of this system is to give the possibility of further education to every citizen who has an internet connection.

4. CONCLUSION

In this paper we presented first two versions of our CALL system for learning Croatian orthography concerning the *yat* reflex – the biggest problem in Croatian orthography. We presented also the third version of the system – our first bigger step towards an intelligent system. One of the essential parts of our further development is the visibility plan. To be able to continue developing our system, it is of greatest importance to gather big amounts of data during the system usage. With the more detailed user profile which informs us about the user's age, education etc., we will be able to draw conclusions that will enable us making this system more personalized depending on user's profile and his skills shown during the system usage. The data we will gather could also show the necessity for developing more triples of word forms, their respective rules and specific algorithms concerning the user's age.

All data gathered during our previous research and system usage, as well as all conclusions we already did and will draw during our research, will not only help us in refining the system to the level of an intelligent system for learning a segment of Croatian language. It will provide us (as well as the whole scientific community interested in this field) with conclusions that can be applied in disciplines like computer-assisted language learning, intelligent systems in education, methodology of language learning and linguistics in general for any, but especially the Croatian language. This is, namely, the biggest project so far in trying to develop an intelligent CALL application for the Croatian language.

5. REFERENCES

[1] Boras, D,; Grmača, D.; Ljubešić, N. (2005): *Computer-assisted Learning of Croatian Orthography Concerning the Yat Reflex (CAL-COR)*, Proceedings Computers in Education (MIPRO 2005), Rijeka

[2] Brusilovsky, P. (1999): *Adaptive and intelligent technologies for web-based education*, In Rollinger, C.; Peylo, C. (Eds.): *Special issue on intelligent systems and teleteaching*, Künstliche Intelligenz, Vol. 4, Universität Hamburg, Hamburg

[3] Carol Chapelle, C. (2000): CALL in the Year 2000: still in search of research paradigms?, Language Learning & Technology, University of Hawai'i at Manoa, Honolulu

[4] Chambers, A.; Davies, G. (Eds.) (2001): *ICT and Language Learning, A European Perspective*, Language Learning and Language Technology Series, Swets & Zeitlinger Publishers, Lisse, The Netherlands

[5] Chapelle, C. A. (2001): *Computer-applications in second language acquisition: Foundations for teaching, testing and research*, Cambridge University Press, New York
[6] Delaney, K. (2002): *Inside Microsoft® SQL Server™ 2000*, Microsoft Press, Redmond
[7] Jeffrey Richter, J. (2002): *Applied Microsoft .NET Framework Programming*, Microsoft Press, Redmond
[8] Kern, R.; Warschauer, M. (2000): *Theory and practice of network-based language teaching,* In M. Warschauer, M.; Kern R. (Eds.) *Network-based language teaching: Concepts and practice,* Cambridge University Press, New York
[9] Lanham, R. G. (1993): *The electronic word: Democracy, technology, and the arts,* University of Chicago Press, Chicago
[10] Michael Levy, M. (1997): *Computer-Assisted Language Learning: Context and Conceptualization*, Clarendon Paperbacks, Oxford
[11] Papanikolaoum, K.A.; Grigoriadou, M. (2002): *Towards new forms of knowledge communication: The adaptive dimensions of a web-based learning environment*, Computers & Education 39, An International Journal (SSCI journal), U.S.A.
[12] Pennington, M.C. (Ed.) (1996): *The Power of CALL*, Athelstan Publications, Houston, Texas
[13] Warschauer, M. (1996): *Computer-assisted language learning: An introduction*, In Fotos, S. (Ed.): Multimedia language teaching, Logos International, Tokyo
[14] Yeh, S.; Lo J. (2005): *Assessing metacognitive knowledge in web-based CALL: a neural network approach*, Computers & Education 44: An International Journal (SSCI journal), U.S.A.
[15] Young, M.; Young, C. (2003): *Deploying Solutions with .NET Enterprise Server*, Wiley Publishing Inc., New Jersey