MIRTO: A NEW APPROACH FOR CALL SYSTEMS

Georges Antoniadis, Sandra Echinard, Olivier Kraif, Thomas Lebarbé, Mathieu Loiseau, Claude Ponton

LIDILEM – Université Stendhal – GRENOBLE 3 – France
{Antoniadis, Echinard, Kraif, Lebarbe, Loiseau, Ponton}@u-grenoble3.fr

Abstract: The MIRTO project aims at designing a pedagogical plateform using Natural Language Processing (NLP) technologies, meant to be used by language teachers. More than an element of quality, the NLP is prerequisite, as our own, for the language learning softwares to be able to teach language as such. MIRTO tries to work out this approach, while offering an NLP based authoring system to create pedagogical objects, such as activities and scenarios.

Key words: E-Learning, Learning Management Systems, NLP for CALL

1. INTRODUCTION

It is generally reckoned that computer science can be a great help for language learning; the fact is that language didacticians and computer scientists do not admit the same acceptation of the term "language". The first consider it as a system of concepts, the latter as a system of forms. Such a difference can be easily explained by the fact that computer science can only process the forms of a language when didactics consider the form as the

materialisation of the concepts. Such a duality is visible in the great majority of language learning software and explains the numerous imperfections. That is why the first part of this short paper will present the main limits of current CALL (Computer Assisted Language Learning) systems. The second part will consist in evoking what NLP (Natural Language Processing) technologies can bring to CALL systems while introducing the philosophy of the MIRTO project. Then, we will present its general structure and the related concepts. The fourth part will illustrate how NLP tools can improve CALL's limits through examples from the prototypal modules developed for MIRTO. Such an illustration will finally allow us to present the future developments and the associated perspectives.

2. THE LIMITS OF CALL SYSTEMS

Since the outset of Computer Assisted Learning in the seventies, specific CALL products have significantly evolved [POT, 2000]. These evolutions are mainly due to the ones in language didactics and in computer science with, for instance, the development of multimedia and networks. However, three recurrent problems are to be noticed [ANT, 2004].

We can first consider the poorness of meaning associated to any linguistic sequence whatever its length, might it be produced by the software and directed to the student or the other way round. At the production as well as the reception of a linguistic sequence, software applications generally consider only a single predetermined aspect of the latter. Effectively, words, syntagms (phrases), sentences and texts are still processed as single character strings since computer science uses numerical form to represent an original form. So, morphosyntactic and semantic language features are not taken into account. This involves an incapacity for these applications to take language specific properties into consideration, which limits the interest, the possibilities and the quality of such tools. That is why, for instance, when a sequence as *Mary is the girl you met last week* is produced to illustrate the omission of a relative in English, the only property of the considered sequence is that precise illustration. The fact that this sequence has several other properties, as the fact that Mary is the subject or that met is the past tense of meet for instance, is not considered in the software and cannot be processed.

The second problem stems from the rigidity of CALL software. It limits the examples to a finite and predetermined set and only allows to use a given text in a given activity and prevents them from being interchangeable. Most of the time, this does not result from a pedagogic choice contrary to

language didactics which can provide open and variable learning contexts as well as adapted exercises for a learner. Computer science would not be able to transform this didactic set of problems in programs without using other kinds of knowledge and competence.

Finally, the third problem concerns the fact that software applications are computer sciences oriented. This orientation forces the language teachers with no or little computing skills to manipulate concepts which do not belong to their set of problems. Thus, instead of expressing pedagogic answers, they are constrained to look for computerized solutions, which connect as much as possible with their own models or pedagogic propositions. Effectively, early CALL systems used to be perfect computerwise but lacked interest on a didactic point of view since it was necessary to have notions in programming in order to design interactive courses.

3. THE MIRTO PROJECT OR THE CALL/NLP COUPLING

In the perspective of covering up these limits, an orientation towards the development of a user-oriented, NLP based, teaching platform appears to be a straightforward answer [ANT, 2002].

Effectively, CALL systems often consider language as given information without being able to take its specific properties into consideration. However, it appears necessary that this material should be processed and then controlled. NLP can bring solutions since its scope of competence stands between linguistics and computer science. It allows to consider the meaning of linguistic form in the way of various applications, but also for language teaching since it offers the possibility of using corpora, detecting and analyzing errors or diversifying learners' pedagogical course.

We lay stress on a real consideration of the different disciplines involved which are as many necessary conditions to overcome the present limits of CALL systems.

3.1 Natural Language Processing

The set of NLP problems can be summarized with two theoretical goals: automatic analysis and automatic generation [BOU, 1998]. Automatic analysis aims at formally representing, by successive parses, the meaning of a written or spoken form. The one of the generation is to automatically produce texts in one particular language or speech with a formal representation of an informative content. In order to reach these goals, parses

are made on four levels which sometimes overlap each other: lexical, morphosyntactic, semantic and pragmatic [FUC, 1993]. The results of each level are obtained by the use of programs. Those programs use methods which exploit forms and associated meaning properties. We can notice that it is not computer science methods which are used but linguistic ones. Even if NLP goals are not reached, some results, as the lexical and morphosyntactic ones, are sufficiently significant and exploitable. Nowadays, each application dealing with human-machine communication is, or will be, concerned with NLP results since every communication has to use language and natural language since it is the most appropriate language for a human. Thus, it appears paradoxical that NLP technologies have been forgotten in most CALL systems, but a few such as ALEXIA [CHA, 2000] or ELEONORE [CHA, 1995], since CALL manipulates language. But as for us, this relative absence is mainly due to a lack of knowledge about NLP for CALL since it can provide a very favourable efficiency-gain-to-additional-cost ratio.

3.2 Use of NLP technologies for CALL systems

The task of NLP is dual: it means an ability to formally determine and represent the informative content (meaning) of whatever language sequence and the possibility to convert at least one sequence from informative content to the formal representation. Those two processes suppose an ability to handle and exploit four kinds of data without associated knowledge: series of characters (morphemes) and their linguistic properties. As we have already noticed, NLP technologies offer satisfying results on characters and words parsing, and great progress concerning sentences.

Hence, concerning the advantages of NLP technologies to cover up the limits of CALL systems [SEG, 2002], several aspects of such a pairing seem conceivable. We can notice that the detection and extraction of words of a text with morphosyntactical criteria may lead to automatical generation of gap-filling exercises. The latter may use the error detection and analysis with "intelligent" feedback. Such an analysis can bring a qualitative exploitation of the learner's course, thus more adaptation and diversification.

The MIRTO project aims at designing a platform for teaching languages and subjects using linguistic material in order to solve some limits of CALL software. It puts its approach into practise while developing a user-oriented NLP based teaching platform and offering new perspectives for CALL systems.

Language teachers are allowed to query for texts or parts of texts according to pedagogical criteria. According to this hypothesis, those texts are to be the basis of their activities or scenarios.

The MIRTO project is determinedly pluridisciplinary, and aims at giving an NLP toolbox to language teachers in order to design scenarios in their own pedagogical set of problems.

4. GENERAL STRUCTURE OF MIRTO

The main goal of MIRTO is to propose to the language teacher the possibility of designing pedagogical scenarios while fully taking advantage of NLP technologies in a user-friendly manner. Thus, those scenarios will be open (dynamical text database), will allow an individualized adaptation according to the learner (automated generation of exercises, qualitative evaluation of the answers...) and should allow new possibilities (work on long texts, automated production of aids or exercises, design of non-linear scenario ...). The approach of MIRTO is determinedly user-oriented since it is meant for language teachers who, a priori have little or no skill in NLP nor in computing. The technical nature of NLP has to be transparent to the language teacher and only the didactic aspects are to be visible and available to him.

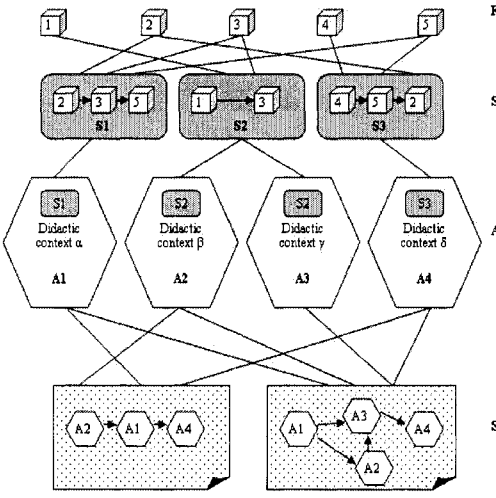

Figure 1 - MIRTO Levels

In that way, four hierarchical levels (function, script, activity and scenario), associated with the text database, structure MIRTO as it is illustrated on figure 1.

4.1 Function level

The functions represent the lower level MIRTO objects. They correspond to a basic NLP process such as tokenization (text splitting in forms) or language identification. Considering its technical nature and its

independence from a didactic application, this level is not visible for the users.

4.2 Script level

This level corresponds to the application of NLP functions to language didactics. A script is a series of functions with a didactic objective.

For instance, the automated design of a gap-filling exercise is considered as a script because it connects the functions of language identification, tokenization, morphological analysis and gap creation depending on parameters chosen by the user.

The working of a script is hidden to the teacher-designer of scenarios since it is presented as a toolbox.

4.3 Activity level

This level with the next one (scenario level) is the didactic core of MIRTO. An activity corresponds to the didactic contextualization of a script (previous level). Its goal (figure 2) is to associate a script with a text from the corpus database, an instruction, possible aids and a facultative evaluation system.

In order to create a gap-filling exercise, one only has to choose to apply the script of the previous example to a text while specifying the gaps criteria (for instance, hiding the preterit verbs and replacing them by their infinitive form), associating an instruction as "Fill in the blank with the preterit form" and specifying the evaluation form of the activity.

The activities definition is realized by the teacher through an adapted authoring system.

4.4 Scenario level

This level allows the teachers to define the sequence of activities in order to answer to their pedagogical objectives throughout the learner progression. This expected progression is not the same for each learner. Effectively, each of them will have a personal learning process linked to different factors. MIRTO is dealing with that reality while proposing non-linear scenario creation. The path through the scenario depends on the individual process of each learner (learning course, evaluation...). That course is stored in a learners' tracing database. For instance, according to his progress in a given scenario, a learner can be redirected to remediation activities, or retry an activity on another text or simply advance in the scenario.

A MIRTO first prototype is being developed right now. The first two levels (function and script) are achieved and the two others are the goal of a current work. The next part presents this realization through examples of activities and scenarios already developed.

5. THE MIRTO PROTOTYPE

The MIRTO prototype may be described as two main modules: the NLP module and the didactic module. The first module allows an NLP specialist to create scripts. By using these scripts, the didactic module offers to language teachers an authoring system in order to create both activities and scenarios.

5.1 The NLP module

In order to design this module, we needed a system that would both be able to integrate NLP tools and to associate them. We are providing a graphical environment in order to create scripts. This interface allows to consult all available functions within the platform. The aggregation of those different functions will constitute the script. The chaining of scripts is weakly constrained. Once validated, the script is stored in a database, so that it can be reused.

Our interface allows us to easily integrate NLP tools, to test them, to connect them (when their outputs are compatible) and then to elaborate MIRTO scripts. However, these script creations must be the result of a close collaboration between a didactic expert and an NLP specialist. Effectively, a script expresses a didactic goal (cf. §4.2).

5.2 The didactic module

For our first prototype, we have chosen to develop mainly a module allowing to show the NLP potential for CALL. This module deals with the two main didactic MIRTO levels: activity and scenario.

5.2.1 The activity design module

This module can be represented as an authoring system with a script toolbox where one can create activities as gap-filling exercises for instance (cf. figure 2), but where the result is managed through a MIRTO script and where each word is associated with linguistic features.

We are able to generate activities using NLP based tools so as to give a meaning to a sequence, without requiring the designer to be a computer scientist. Effectively, simple NLP applications can add a lot of functions to the activities. A set of generation rules, that is to say the script, allows to determine the form and the grammatical features of the words that are to be removed, and the information that are to be shown in the gaps. These kinds of activities allow the learner to have access to additional information such as a comprehension assistant: grammatical information or links to external resources (hypertextual links) which are automatically added. Moreover, these activities can offer a lot of advantages on evaluation, which gives the possibility of creating non-linear scenarios or advanced feedbacks.

Activities and their linguistic features are stored in a database. Thus, they can be easily modified or shared. Those NLP based activities allow to suggest potential answers to the language teacher. However, in case of ambiguities, the activity designer may be able to modify the "system answers". He will have access to both answer handling and test modes, and will be able to modify activities by removing a "gap" for instance.

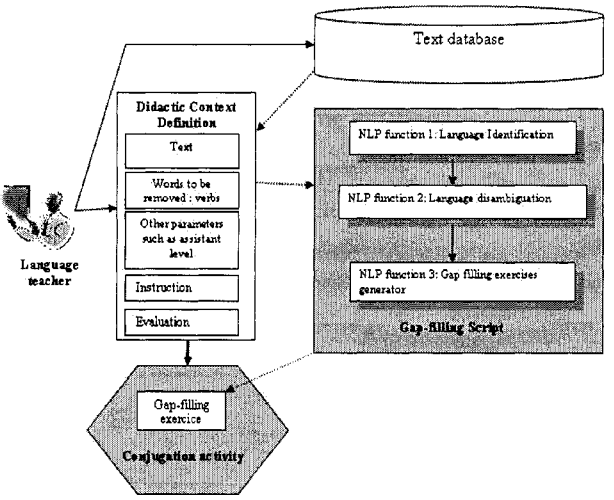

Figure 2 - Use of MIRTO scripts to design a didactic activity

So, he can adapt and diversify his activities, which can be also possible through an existing scenario.

5.2.2 The scenario design module

Another aspect of the didactic module is the creation of pedagogical scenarios. We have chosen, similarly to activities, to conceive a user-oriented scenario design interface. We have then developed a user-oriented Java prototypal applet in order to design tree structured scenarios. The pedagogical designer will choose activities objects and will join them with a simple arrow. He will be able to devise the pedagogical course process of the

activities. The configuration of transition from an activity to another is being developed as well as a "tracing" module, but they will depend on NLP error analysis possibilities and on the kind of activities. As for the activities, scenarios involve much flexibility and the language teacher could imagine and design scenarios which will be easily adaptable and variable.

6. PERSPECTIVES

After presenting what is already developed, it is necessary to stress on perspectives and future developments. The first one is the integration of an answer analysis module. We are working on a learner corpora analysis in order to define a reliable error analyzer involving advanced evaluation. It will allow to improve the non-linear scenario design and will give the possibility to individualize further learner courses, thus providing a great improvement for the learner's production. Effectively, such an answer analyzer must be able to locate orthographical errors as well as the non-acquisition of a grammatical notion.

We are now developing the "tracing module" in order to manage with non-linear scenarios and the visualization of the learners'course : the effective work, the collection of errors, their analysis, the aids used or the duration. The tracing will allow the teacher to access to what is acquired or not. He will be able to adapt his lessons in presential classes or by the help of others MIRTO scenarios depending on the entire group results.

Furthermore, we are concurrently working on the design of a "pedagogically indexed text-database" [LOI, 2003]. This database should allow teachers to find texts according to pedagogical criteria such as what activities to perform with the text. Once available, it should be integrated within the MIRTO platform, thus broadening the perspectives in terms of activity authoring. Such a database should allow them to query for texts or parts of texts according to pedagogical criteria. For instance, an English teacher should ideally be able to query for a text to introduce the present simple in contrast with the present continuous to a class of French beginners. The teachers should be able to add their own texts to the database in order to share them with peers and benefit from each other's experience.

In the long run, we consider to integrate MIRTO in a pedagogical platform which would allow to manage with pedagogical resources organization, the users and collaboration tools and to propose the possibility of designing non-NLP based activities. We go towards a pedagogical authoring system based on the NLP possibilities and on the didactic set of problems of language learner.

REFERENCES

[ANT, 2004] Antoniadis G. 2004. "Les logiciels d'apprentissage des langues peuvent-ils ignorer le traitement automatique de la langue ?", Les cahiers de l'APLIUT, 23,2.

[ANT, 2003] Antoniadis, G. 2003. "Le TAL : une composante centrale pour les outils d'apprentissage des langues." Conférence invitée, université de Medellin, 22 mai 2003, Medellin, Colombie.

[ANT, 2002] Antoniadis, G. & Ponton, C. 2002, "Le TAL : une nouvelle voie pour l'apprentissage des langues." Communication au colloque UNTELE'2002, 28-30 mars 2002, Compiègne.

[BOU, 1998] Bouillon, P. 1998. "Traitement automatique des langues naturelles", Champs linguistiques Universités francophones, Paris.

[BOU, 2001] Bourda, Y. 2001. "Objets pédagogiques, vous avez dit objets pédagogiques ?", in Cahier de Gutenberg, GUT2001, 39-40, Metz, France.

[CHA, 1998] Chanier, T. 1998. "Relations entre le TAL et l'ALAO ou l'ALAO un "simple" domaine d'application du TAL ?" Communication: International conference on natural language processing and industrial application (NLP+IA'98). août 1998, Moncton, Canada.

[CHA, 1995] Chanier, T. & Renié, D. 1995. "Collaboration and computer-assisted acquisition of a second language" Computer-Assisted Language Learning 8, 1: 3-30.

[CHA, 2000] Chanier, T. & Selva, T. 2000. "Génération automatique d'activités lexicales dans le système ALEXIA" Sciences et Techniques Educatives 7, 2 : 385-412. Paris : Hermes

[FUC, 1993] Fuchs, C., Danlos, L., Lacheret, A., Luzzati, D. & Victorri, B. 1993. "Linguistique et Traitements automatiques des Langues". Paris : Hachette.

[HAB, 1998] Habert, B., Nazarenko, A. & Salem, A. 1998. "Les linguistiques de corpus" Paris: Armand Colin.

[LAS, 1998] Lassila, O., 1998, "Web Metadata : A matter of semantics" IEEE Internet Computing, July-August 1998, 30-37.

[LOI, 2003] Loiseau, M. 2003. "Vers la création d'une base de données de ressources textuelles indexée pédagogiquement pour l'enseignement des langues" Mémoire de DEA Sciences du Langage, Université Stendhal, Grenoble.

[NER & al., 1998] Nerbonne, J., Dokter, D. & Smit, P. 1998. "Morphological Processing and Computer-Assisted Language Learning.". Computer-Assisted Language Learning 11, 5 : 543-559.

[POT, 2000] Pothier, M. & Chanier, T. (éditeurs). 2000. Numéro spécial Eurocall'99. Revue Apprentissage des Langues et Systèmes d'Information et de Communication (ALSIC). vol.3, 1, juin.

[SEG, 2002] Segond, F (éditeur). 2002. "Les outils de TAL au service de la e-formation en langues." Multilinguisme et traitement de l'information : 223-250. Paris : Hermes

[VAN, 2003] VANDEVENTER FALTIN, A. 2003. «Natural language processing tools for computer assisted language learning». Linguistik online 17, 5/03.

[YAN, 1997] YANG, JC. & AKAHORI, K. 1997. "Development of computer assisted language learning system for Japanese writing using natural language processing techniques: A study on passive voice". Intelligent Educational Systems on the World Wide Web, 8th World Conference of the AIED Society, Japan.