ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques

Itziar Aldabe[1], Maddalen Lopez de Lacalle[1], Montse Maritxalar[1],
Edurne Martinez[2], and Larraitz Uria[1]

[1] Department of Computer Languages and Systems,
Computer Engineering Faculty, University of the Basque Country
P.O. box 649, E-20080 Donostia, Spain
{jibalari,mlopezdelaca002,montse.maritxalar,larraitz}@ehu.es
[2] Computer Science Department, Udako Euskal Unibertsitatea (UEU)
ELEKA Language Engineering Enterprise
edurne@eleka.net

Abstract. Knowledge construction is expensive for Computer Assisted Assessment. When setting exercise questions, teachers use Test Makers to construct Question Banks. The addition of Automatic Generation to assessment applications decreases the time spent on constructing examination papers. In this article, we present ArikIturri, an Automatic Question Generator for Basque language test questions, which is independent from the test assessment application that uses it. The information source for this question generator consists of linguistically analysed real corpora, represented in XML mark-up language. ArikIturri makes use of NLP tools. The influence of the robustness of those tools and the used corpora is highlighted in the article. We have proved the viability of ArikIturri when constructing fill-in-the-blank, word formation, multiple choice, and error correction question types. In the evaluation of this automatic generator, we have obtained positive results as regards the generation process and its usefulness.

1 Introduction

Nowadays, it is widely recognized that test construction is really time-consuming and expensive for teachers. The use of Computer Assisted Assessment reduces considerably the time spent by teachers on constructing examination papers [11]. More specifically, e-assessment helps teachers in the task of setting tests. For example, in the eLearning Place [3] learning providers create the question bank by means of a TestMaker, a Java Virtual Machine tool. The manual construction of questions is also a fact in SIETTE[5], a web-based tool for adaptive testing. TOKA[8], a web-application for Computer Assisted Assessment, provides teachers with a platform for guided assessment in which, in addition, they construct exercises. All these tools have been used for the assessment of different subjects. However, the work we present in this article is focused on language learning. In our case, learning providers do not have to waste time preparing the questions of the exercises since they are automatically generated.

M. Ikeda, K. Ashley, and T.-W. Chan (Eds.): ITS 2006, LNCS 4053, pp. 584 – 594, 2006.

Some research on automatic generation of questions for language learning has been recently carried out. [12] includes a maker for generating question-answer exercises. [9] reports on an automatic generation tool for *Fill-in-the-Blank Questions* (FBQ) for Italian. [13] describes a method for automatic generation of *multiple choice* questions together with the Item Response Theory based testing to measure English proficiency. [10] uses different techniques such as term extraction and parsing for the same purpose. [4] also describes how to apply Natural Language Processing (NLP) techniques to create *multiple choice* cloze items. Finally, [7] reports a machine learning approach for the automatic generation of such type of questions. In the mentioned works, the authors present different methods for the automatic generation of language questions based on NLP techniques. However, in almost all of them the methods and the system architectures are only focused on a single question type. In contrast, this article proposes a NLP based system, which is able to generate four different types of questions: *FBQ, word formation, multiple choice,* and *error correction*. Moreover, we propose a system where all the inputs and outputs are in XML markup-language. Concerning to its architecture, we point out some existing differences among our system and the above mentioned ones in section four.

The system we propose provides teachers and testers with a method that reduces time and expenditure for testing Basque learners' proficiency. The user chooses which linguistic phenomena s/he wants to study, and what types of questions s/he wants to create. The system generates, automatically, the question types that the user has requested. For this, the system makes use of a real corpus and some NLP tools for Basque developed in the IXA research group[1].

In section two, we present ArikIturri, the automatic corpus-based question generator. In section three we briefly describe the question model used by this system. Section four deals with the development of the system; we talk about its architecture, as well as the use of the NLP tools and the source corpus. In section five, we comment on the evaluation of ArikIturri, and we present Makulu, the assessment application we have used for the evaluation of the generator. Finally, some conclusions and future work are outlined.

2 ArikIturri: The Question Generator

Here we present a question generator for Basque, named ArikIturri. This is a system with an open architecture (section four) to generate different types of questions. ArikIturri makes use of a data bank, which consists of morphologically and syntactically analysed sentences where phrase chunks are identified. The input of ArikIturri is represented by XML mark-up language. The outputs are question instances of a model defined also in XML mark-up language (see figure 1).

The system generates automatically the question instances. For that, it makes use of two kinds of language resources: NLP tools and specific linguistic information for question generation.

ArikIturri is independent from the assessment application, which will use the questions created by the generator. Indeed, it is the assessment application that determines

[1] http://ixa.si.ehu.es/Ixa

the type of the questions to be generated as well as the linguistic phenomena treated in those questions. For this reason, the question instances generated automatically by ArikIturri must be imported to the assessment application. This way, as showed in figure 1, different assessment applications can benefit from the question generator.

Although the knowledge representation is not a matter of this paper, we want to outline that the importation of question instances implies the matching between the concepts defined in the question model of ArikIturri and the representation of the domain of the assessment application that is using the automatic question generator.

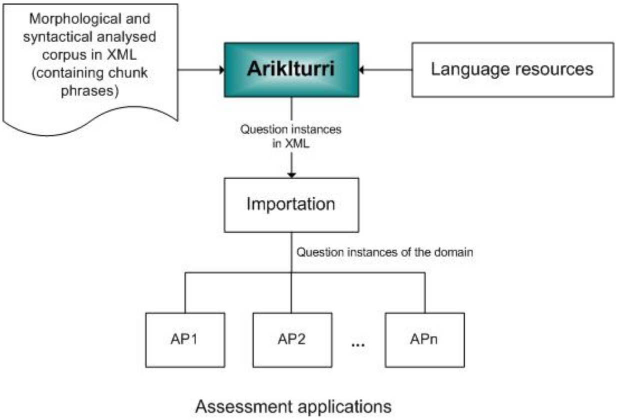

Morphological and syntactical analysed corpus in XML (containing chunk phrases)
ArikIturri
Language resources
Question instances in XML
Importation
Question instances of the domain
AP1
AP2
...
APn
Assessment applications

Fig. 1. ArikIturri is independent from the assessment applications

As said before, ArikIturri can generate different types of questions: *FBQ, word formation, multiple choice,* and *error correction.*

As concerns FBQ, when deciding the blanks of the question, the generator itself chooses which are the units to be removed from the text. In our approach, the system can construct questions with more than one blank, and each of them could be filled with one or more words, depending on the exercise. In order to get the blanks, the system identifies the morphosyntactic categories of the phrases in the source sentence. So far, we have experimented with two types of linguistic phenomena: morphological inflections and verb conjugation for auxiliary and synthetic forms.

Word formation questions consist of a given sentence and a word whose form must be changed in order to fit it into the sentence. To generate this question type, we use a lemmatiser for Basque [2], which gives the word (a lemma) to be changed in the blank.

In error correction questions, the aim is to correct the errors, which can be marked or not. And, in multiple choice questions types, we find a set of possible answers. Only one of them could be correct in that context, whilst the rest are incorrect answers, i.e. distractors.

In fact, in the case of multiple choice and error correction question types, the generator has to create distractors, which are the incorrect answers the system offers. The techniques used for the automatic generation of distractors can have a big influence in the results. In our case, we have defined different methods to generate distractors. As far as the generation of inflection forms is concerned, the system uses techniques of replacement and duplication of declension cases (inessive, ablative, dative…), number (singular/plural) or the inflection paradigm (finite, indefinite). As regards the verb forms' generation, the system creates the distractors by changing the subject person, the object person, the verb mode, the tense, the aspect or the verb paradigm. We have defined these techniques to create distractors as a parameter of the generator. This way, we are able to research and improve the generation of distractors depending on the results we obtain.

We also want to say that, although test correction in assessment applications for learners is not a matter of our present research, we think it is important to consider this aspect too. For the moment, and based on the source corpus, the generator provides us with only one possible correct answer in each question instance.

3 A Brief Description of the Question Model

As we have already said, the outputs of the generator are question instances of a question model defined in the XML mark-up language. A question is not an isolated concept but it is represented as a part of a whole text.

In this model, a question can have more than one answer focus. For example, FBQ can have more than one blank to fill in. Concretely, a question could have as many answer focuses as phrases are identified in the question. Of course, if the number of answer focuses was equal to the number of phrases, we would have a completely empty question with no sense. That means that the generator has to control the number of answer focuses in respect to the number of phrases of the source sentence.

Therefore, one of the main characteristics of the question model we have defined is that we find different answer focuses within the same question, i.e. more than one possible blank in a FBQ exercise. The focus is always a phrase of the sentence. The phrase, on its own, has a head, which corresponds to the words of the blank of the generated question. The head concept is defined by one possible answer, i.e. the one corresponding to the source text, any number of distractors (zero in the case of some question types), and lexical and morphosyntactic information.

Finally, it is important to stand out that the order of the phrases in the generated questions is not necessarily the same as the order in the source sentence. This is a very enriching characteristic of this question model because Basque is a free word order language. Besides, the fact that the phrases have not to be in a fixed position offers us the chance to extend the application of the method to generate new exercise types such as word order, transformation and question answering.

4 Development of the System

In this section, we explain the architecture of ArikIturri and some important aspects studied during the development of its architecture. Indeed, the source corpus and the NLP tools that the generator uses have a big influence in the quality of the system. Basically, it is important to have robust NLP tools that provide us with correct linguistic analysis. Those tools have been developed in the IXA research group at the University of the Basque Country.

4.1 The Architecture

Here we describe the main modules of ArikIturri's architecture. The generator uses as input a set of morphologically and syntactically analysed sentences (the tagged corpus), represented in the XML mark-up language, and it transforms them into the generated questions, represented in XML. The representation of the results obtained in the intermediate stages is also in XML.

As mentioned in the introduction, there are some differences among the architecture of this system and the architectures of previous works ([9],[12]). We have distinguished an *Answer Focus Identificator* module and an *Ill-formed questions rejecter* module. [13] also includes a module to reject questions, which is based on the web.

Figure 2 shows the main modules of the architecture. Depending on the parameters' specifications, *the sentence retriever* selects candidate sentences from the source tagged corpus. In a first step, it selects the sentences where the specified linguistic phenomena appear. Then, the *candidates selector* studies the percentages of the candidates in order to make random selection of sentences depending on the number of questions specified in the input parameters.

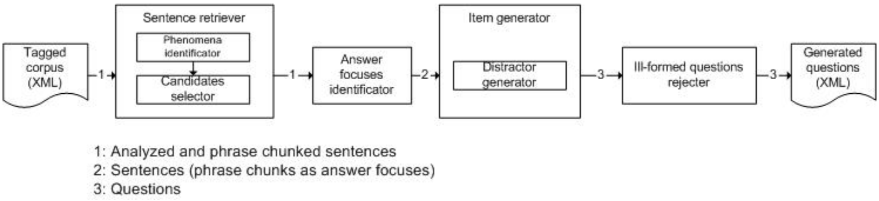

Tagged corpus (XML)
1
Sentence retriever
Phenomena identificator
Candidates selector
1
Answer focuses identificator
2
Item generator
Distractor generator
3
Ill-formed questions rejecter
3
Generated questions (XML)
1: Analyzed and phrase chunked sentences
2: Sentences (phrase chunks as answer focuses)
3: Questions

Fig. 2. The main modules

Once the sentences are selected, *the answer focuses identificator* marks out some of the chunked phrases as answer focuses depending on the morphosyntactic information of the phrases. Then, the *item generator* creates the questions depending on the specified exercise type. That is why this module contains the *distractor generator* submodule. At this moment, the system has already constructed the question instances. However, as the whole process is automatic, it is probably that some questions are ill-formed. Because of that, we have included *the ill-formed questions rejecter* in the architecture. In section 4.2 we explain how the main modules work, from the perspective of the NLP tools.

4.2 The Influence of the NLP Tools and the Source Corpus

As we foresaw at the beginning of our research, our experiments have proved that the source corpus and the NLP techniques used in the process of question generation determine the quality of the obtained questions. In the next lines, we explain the influence of these two aspects in the development of the system.

4.2.1 The NLP Tools

The results of the question generation based on corpora depend very much on the matching between the linguistic information of the answer focuses of the question and the specific linguistic phenomena that teachers want to test. When working with NLP tools, the robustness of those tools undoubtedly determines the results. The results we have obtained depend, in some way, on the quality and sophistication of the morphosyntactic parser, the syntactic parser and the phrase chunker for Basque [1] we have made use of for the analysis of the real corpus. In some small experiments we carried out when developing ArikIturri, we studied the output of the mentioned NLP tools. Since the information given in this output did not always respond to our needs, we realised that those tools determine the results of *the sentence retriever.* As a consequence, we had to discard the study of some linguistic phenomena, such as the inflection of demonstrative pronouns or the genitive case. This problem could be solved, of course, by making some changes in the mentioned NLP tools.

The task of the *answer focuses identificator* depends on the quality of the chunked phrases. In our implementation, if the phrase has different analysis corresponding to different linguistic phenomena, the focuses' identificator do not consider that phrase as a candidate answer focus.

We have adapted the verb conjugation tool and the morphological declension tool for Basque language [6] in order to generate different well-formed words as distractors that are incorrect in a particular context, i.e. in the generated question. Sometimes, the conjugation tool and the declension tool give no output because its input parameters, automatically set by the *distractor generator,* have no sense. In these cases, the declension tool does not produce any distractor. This way, if the number of generated distractors does not match with the input parameters of ArikIturri, the *ill-formed distractors rejecter* will mark the generated question as deleted. The rejecter also controls if there are duplicated distractors in the same answer focus. This is possible, for example, because Basque sometimes uses equal inflection forms for different linguistic features.

Here we show an example of a rejected multiple choice question type. The module rejects the question because there are two identical distractors i.e. b) and c) for different inflection forms[2]. The choices of the question are *answer focuses* where the **head** of the answer is in bold.

"Dokumentua sinatu zuten________"
("They signed the document ________")

a) *alderdiaren **izenean*** (innesive definite singular – in the name of the political party)
b) *alderdiaren **izenetan*** (innesive definite plural – in the name of the political parties)
c) *alderdiaren **izenetan*** (innesive indefinite – in the name of certain political parties)
d) *alderdiaren **izen*** (lemma –name of the political party)

In respect to the word formation questions, the results make sense if some variation of the showed word matches with the answer focus. In our implementation, the word is an automatically identified lemma. That is why the correctness of the lemmatiser used when disambiguating different lemma candidates for the answer focus

[2] In this paper, we show the example as presented to the expert teacher, after the importation of the XML instances.

considerably affects the evaluation of the appropriateness of the word formation question type.

4.2.2 The Source Corpus

The language level of a text is a controversial aspect because it is difficult to define it. However, language schools are used to classify real texts according to the established language levels. In order to carry out our experiments, we have analysed the corpora classified into three different language levels. Expert teachers chose the texts. Initially, we thought that the number of instances for each linguistic phenomenon would change depending on the language level. However, the results of the analysed texts show that there is not significant difference on the rates, at least as far as morphological inflection and verb conjugation are concerned. Based on these results, we were not sure about the importance of the distinction by language levels for our experiments. And we finally decided to use only one level corpus, i.e. the high language level corpus (234 texts) for making experiments in order to define which linguistic phenomena we would use in the evaluation (section five). Using high language level corpus, we have avoided the noise that teachers would generate when discarding questions at lower levels because of the difficulty the students could find to understand the sentences.

The linguistic phenomena defined in the curricula of Basque language schools were the starting point of our experiments. Initially, we analysed the whole corpus (1303 texts - 44009 sentences) and found out that some of the linguistic phenomena taught at the language schools did not appear in that corpus. Concretely, the results of the experiments made with the *sentence retriever* showed that the number of appearances of certain verb forms was too low in the corpus. This way, we verified that the corpus limits the linguistic phenomena that can be treated in the generated questions. Moreover, the percentage of appearance of the phenomena in the corpus did not match with the importance rate that teachers gave to the learning of these linguistic contents.

As we said in section two, we focused our experiments on the study of morphological inflections and verb conjugation. With the sentence retriever, we used a sample of the corpus, i.e texts of high language level (234 texts; 10079 sentences). The results of the experiments with the *sentence retriever* determined the criteria for the evaluation of the system. More specifically, we chose five different inflection cases (sociative, inessive, dative, absolutive and ergative) and four different verb forms (present indicative, past indicative, present indicative-absolutive, present indicative-ergative) corresponding to different paradigms, modes, aspects and tenses. The sample corpus contained 16108 instances[3] of the selected inflection cases (703, 4152, 1148, 7884 and 2221, respectively) and 7954 instances of the selected verb forms (3814, 100, 2933 and 1107, respectively).

5 The Evaluation of ArikIturri

The experiments we have carried out in order to generate questions automatically have proved the viability of ArikIturri when constructing fill-in-the-blank, word

[3] 16108 inflection cases detected in 10079 sentences means that some of the cases appear more than once in the same sentence.

formation, multiple choice, and error correction questions. Although we have generated four different types of questions, for the evaluation of the system we have only taken into account the results obtained with multiple choice and error correction question types.

As said before, the NLP tools and the source corpus determine, in some way, the results of the question generator. As a consequence of the experiments carried out during the development of the system, we decided to evaluate ArikIturri taking as source data the high language level corpus and focusing on some specific linguistic phenomena, i.e. five inflection cases and four verb forms.

In this section, we present a manual evaluation of the questions created by our automatic generator. For this evaluation, we have used Makulu, a web-application developed for Computer Assisted Assessment. In this way, we demonstrate that the question instances generated by ArikIturri have been imported to the domain of the Makulu assessment application (see figure 1).

5.1 The Selected Corpus

Taking into account the few economic and human resources we had to carry out a significant evaluation, we decided to use a corpus of 1700 sentences selected from the high language level corpus. As said in section 4.2, this corpus consisted of 10079 sentences. The sentence retriever identified the selected linguistic phenomena for the evaluation, and, out of 10079 sentences we chose 1700 at random.

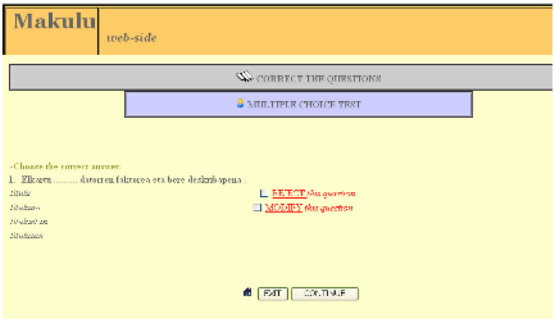
As the study of the automatic generation of distractors was also an interesting aspect, we limited our evaluation to multiple choice (500 sentences) and error correction (1200 sentences) question types. The reason for selecting a higher number of error correction questions was that the evaluation of this type of questions is less time-consuming.

Once we selected the analysed sample corpus, we used ArikIturri to automatically generate the questions. The *ill-formed questions rejecter* of the generator automatically rejected 58 multiple choice question instances and 292 error correction instances out of all the generated questions. This way, we obtained a sample of 1350 question instances to evaluate.

5.2 The Manual Evaluation

For the manual evaluation of ArikIturri, we have used Makulu. Makulu is a web-based assessment application for helping teachers to set test questions and assessing learners in their language proficiency. Makulu groups different questions in order to present a whole exercise. Students have two options: to make the exercises within a whole text, or to have exercises composed of grouped questions. Makulu gives to the students the results of their sessions as well as the correct answers. The task of the teachers is to set the questions that are used in learners' tests.

In order to evaluate the results of ArikIturri, we have used the questions' setting view of Makulu with an expert teacher. Makulu requests ArikIturri to generate language questions. These questions are imported to the Makulu's database. After setting the tests into Makulu, teachers can modify some components of the answer focus of the automatically generated questions. For instance, in multiple choice question type they could change the distractors, but they can never modify the correct answer, because it corresponds to the source corpus. In the next figure we can see an example of an automatically generated question. The teacher has different options: to accept it on its own, to discard it if it is not an appropriate question, or to modify it if s/he considers that, among the options, there is more than one possible correct answer. The set of rejected and modified questions give us a way to evaluate the automatically generated questions.



-Choose the correct answer:

1. Elkarren .... datozen faktorea eta bere deskribapena .

[illegible]

[illegible]

[illegible]

[illegible]

□ REJECT this question

□ MODIFY this question

Fig. 3. Teacher's evaluation of the questions

In the next lines, we comment on the results obtained from the session with the expert language teacher. The teacher actually evaluated 1350 question instances by means of the Makulu web application. She spent 15 hours for manual evaluation. In the evaluation, we asked to the expert teacher to modify or reject questions only if they were not well formed. The expert teacher evaluated 908 questions of error correction type and 442 questions of multiple choice type. If we consider that all the questions discarded or modified by the teacher were not well generated, the results show that the percentage of the accepted questions was of %83,26 in the case of error correction questions and of %82,71 in the case of multiple choice questions. These percentages show us that the developed automatic generator obtains indeed good results. This assertion becomes even more important if we consider the time that the expert teacher needs for setting the questions. It is clear that the setting of the same number of questions with an assessment application of manual construction is more expensive and time-consuming.

Considering that, in the case of error correction questions the generator creates only one distractor per each answer focus (the error), and three different distractors in the case of multiple choice questions, the percentage of well-formed questions should be higher for error correction questions. In addition, a deeper study of the results shows that the methods used for generating distractors and the linguistic phenomena seem to have a big influence on the correctness of the generated questions. These aspects imply a more exhaustive evaluation of ArikIturri.

6 Conclusions and Future Work

The automatic generation of knowledge construction reduces considerably the time spent by teachers on constructing exercises. In this paper, we have shown the results obtained in the development of a system, ArikIturri, for automatic generation of language questions. Concretely, we have proved the viability of this system when constructing, automatically, fill-in-the-blank, word formation, multiple choice, and error correction question types.

The experiments carried out during the implementation of the system have proved that the source corpus and the NLP techniques used in the process of question generation determine the quality of the obtained questions.

ArikIturri is independent from the assessment application, which will use the questions created by the generator. Indeed, it is the assessment application that determines the type of the questions to be generated, as well as the linguistic phenomena treated in those questions. In the present work, we have experimented with Makulu, an assessment application for helping teachers to set test questions and assessing learners in their language proficiency. We have used the questions' setting view of Makulu, in order to evaluate the results of ArikIturri.

We have also presented the results of the evaluation of ArikIturri with multiple choice and error correction question types. Those results demonstrate that the automatic generator is good. In fact, the well-formed questions are more than %80. Moreover, the use of this generator is less expensive and time-consuming than manual construction of language questions. It would be interesting to evaluate the required time to create the same number of questions without ArikIturri; in this way, we would have a real precision of the time expert teachers can save with the system.

For the near future, we foresee to carry out deeper evaluations for studying the quality of the methods used to generate distractors. We also consider very important to make new evaluations with language learners and to compare the results given by different expert teachers. In addition, we are planning to make new experiments to generate new types of test questions such as question answering, word order and transformation.

References

1. Aduriz, I., Aranzabe, M., Arriola, J., Díaz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.,:A Cascaded Syntactic Analyser for Basque. Computational Linguistics and Intelligent Text Processing. 2945 LNCS Series. Springer Verlag. Berlin. (2004) 124-135
2. Alegria, I., Aranzabe, M., Ezeiza, A., Ezeiza, N., Urizar, R.: Robustness and customisation in an analyser/lemmatiser for Basque. LREC-2002 Customizing knowledge in NLP applications workshop (2002)
3. Boyle, A., Russell, T., Smith, S., Varga-Atkins, T.: The eLearning Place: progress report on a complete system for learning and assessment. Proceedings of the Eight International Computer Assisted Conference. M. Ashby (eds.) http://www.caaconference.com/ (2004) 71-77
4. Chao-Lin Liu, Chun-Hung Wang, Zhao-Ming Gao, Shang-Ming Huang: Applications of Lexical Information for Algorithmically Composing Multiple choice Cloze Iems. Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, Ann Arbor, (2005) 1-8
5. Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J.L., Ríos, A: SIETTE: A Web-Based Tool for Adaptive Testing. International Journal of Artificial Intelligence in Education, 14(1), (2004) 29-61

6. Díaz de Ilarraza, A., Maritxalar, M., Oronoz, M: IDAZKIDE: an intelligent CALL environment for second language acquisition. Proceedings of a one-day conference "Natural Language Processing in Computer-Assisted Language Learning" organised by the Centre for Computational Linguistics , UMIST, in association with EUROCALL, a special ReCALL publication. UK. (1999) 12-19
7. Hoshino, A., Nakagawa, H.: A real-time multiple-choice question generation for language testing. Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, Ann Arbor. (2005) 17-20
8. Kerejeta, M., Larrañaga, M., Rueda, U., Arruarte, A., Elorriaga J.A.:A Computer Assisted Assessment Tool Integrated in a Real Use Context. In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies. ICALT. (2005) 848-852
9. Kraift, O., Antoniadis, G., Echinard, S., Loiseau, M., Lebarbé T., Ponton C.: NLP Tools for CALL: the Simpler, the Better. Proceedings of the InSTIL/ICALL2004 – NLP and Speech Technologies in Advanced Language Learning Systems, Venice. (2004)
10. Mitkov, R., An, L.: Computer-aided generation of multiple-choice tests. Proceedings of the 1st Workshop on Building Educational Applications Using NLP. HLT-NAACL. (2003) 17-22
11. Pollock, M.J., Whittington, C.D., Doughty, G.F.: Evaluating the Costs and Benefits of Changing to CAA. Proceedings of the Fourth International Computer Assisted Conference CAA, http://www.caaconference.com/. (2000)
12. Schwartz, L., Aikawa, T., Pahud, M.: Dynamic Language Learning Tools. Proceedings of the InSTIL/ICALL2004 – NLP and Speech Technologies in Advanced Language Learning Systems, Venice. (2004)
13. Sumita, E., Sugaya, F., Yamamota, S.: Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the Blank Questions. Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, Ann Arbor. (2005) 61-68