Automatic acquisition of didactic resources: generating test-based questions

Itziar Aldabe, Maddalen Lopez de Lacalle, Montse Maritxalar
Dept. of Computer Languages and Systems
University of the Basque Country
20080 Donostia
{ialdabe001,mlopezdelaca002, montse.maritxalar}@ehu.es

Abstract

Natural Language Processing (LNP) Techniques facilitate the automatic acquisition of some didactic resources. Using corpora as source data, Arikiturri, is able to automatically generate test-based questions for Technology Supported Learning Systems.

In this article we define some concepts of the generic question model underlying Arikiturri, and explain an experiment with experts in order to evaluate the quality of the generated questions.

1. Introduction

The need of tools that help on the semi-automatic construction of the domain module of Technology Supported Learning Systems was claimed by Murray [12]. In this way, some research on the use of electronic documents has been already done in order to detect and extract didactic resources from them [15]. The use of Natural Language Processing (LNP) Techniques facilitate this detection and extraction processes. For example, [8] presents a domain independent method based on NLP and heuristic reasoning to acquire the domain module from electronic documents and their indexes. In our proposal, we also use NLP to construct didactic resources from electronic documents. However, our aim is not to detect such resources, but to generate them.

Some models of domain representation have been already developed ([10], [5]). One of our goals is to create a question model where not only stem and distractors are represented but also their corresponding topic information as well as the heuristics used for their generation. The human developers of didactic resources can consult them, and also add information related to their own experience for updating the heuristics. With this purpose, we have developed a post-editing environment where teachers can evaluate the generated questions and adapt them to the necessities specified in the curriculum. In this article, by means of an evaluation, we demonstrate that it is possible to generate quality language exercises from corpora.

Section two starts briefly explaining Arikiturri, a system to create automatically questions. Section three defines some concepts concerning the representation of the questions. Section four describes an experiment with experts to evaluate, using a post-editing environment, the output created by Arikiturri. Finally, some conclusions and future work are outlined.

2. Arikiturri: a system for automatic question generation

Test construction is a time-consuming and expensive task for teachers. However, the use of Computer Assisted Assessment reduces considerably the time teachers spend constructing examination papers [13]. In addition, if questions are automatically generated by means of techniques such as NLP the time decreases [11].

In [1], we presented the architecture of a system named Arikiturri. It was developed for the automatic generation of didactic resources. Arikiturri is specifically focused on automatic question generation from corpora and it is based on NLP tools. Despite we made the experiments with the Basque language, the architecture of the system is independent from the language of the source corpora.

ArikIturri is a system with an open architecture developed to generate different types of questions from educational corpora. By the time, we have experimented with fill-in-the-blank, error correction, multiple-choice, word formation [1] and short answer [2] questions. The input of the system consists of a databank which is composed of morphologically and syntactically analysed sentences where phrase chunks are identified. This input is represented by the XML mark-up language.

The candidate sentences of the questions are automatically extracted from a databank, depending on the topic of the question. However, distractors are automatically generated words; they are not extracted from any databank. Due to the high level of inflection of Basque, it is impossible to store every word form in a dictionary, even in a compressed way. Thus, we use a general purpose morphological generator for the generation of the distractors. By contrast, the heuristics used by the generator are not automatically generated, but they are based on experts' knowledge. We have also done some experiments for the automatic generation of the heuristics, but we don't explain them in this article.

The outputs of the system are question instances of a model also defined in XML. ArikIturri is independent from any assessment application. The application will import the questions created by the generator and will determine the type of such questions as well as the topic to be treated.

3. Representation of the questions

In this section we will define some concepts concerning the representation of the questions. Moreover we present a question model represented in UML.

3.1. What is a question?

In the last years, some research on automatic generation of language questions has been carried out. Among others, reading comprehension, vocabulary, cloze questions and grammar tests are automatically generated. All the researches use different types of NLP techniques and resources, and they are generally focused on the creation of only one type of question.

Depending on the type of questions that those systems generate, different definitions are provided for explaining the *question* concept. For instance, multiple-choice questions are generally defined ([6], [9], [11], [14]) as a stem, the correct answer and the rest of the possible choices (distractors).

Other systems not only generate multiple-choice questions but also another question types. In word bank [3], the tester sees a list of answer choices, followed by a set of questions of statements. In [4], an error detection item consists of a partially underlined sentence where one choice of the underlined part represents the error and the other underlined parts act as distractors.

When we developed ArikIturri, we felt the need to define the concepts concerning the representation of a question in an independent way from the question type. Our objective was to define a model for different types of questions. Moreover, we wanted to create a model where questions as well as the heuristics used in the generation process were collected together. In our model we define a question as a sentence or a clause where the topic the learner has to work on appears. And, a question is not an isolated concept but it is represented as a part of a whole text.

3.2. The question model

With the aim of defining the question model of the generator, we took into account different types of questions: fill-in-the-blank, error correction, multiple-choice, word formation and short answer question types.

In [2] we presented a first version of the model. Figure 1 shows an extension of that question model represented in UML. We define an exercise as a set of questions. The question is composed of three main components: the *topic*, the *answer focus* and the *context*, all of them compulsory attributes. We define the answer focuses as the chunks of the sentence where the topic appears. The rest of the chunks of the sentence are collected into the context. As regards the chunks, the structure of the model does not only store all their words but it also provides their analysis. A question is represented as a part of a whole text and the *pos* attribute refers to the

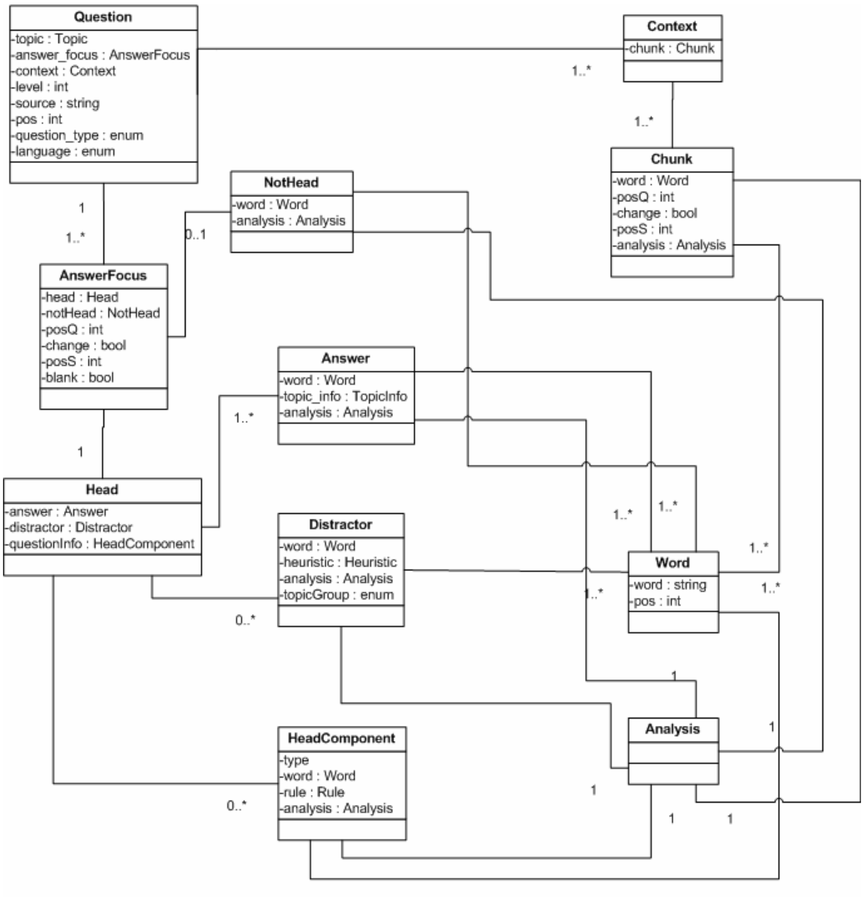


Figure 1 - The question model

position of the sentence into the source text. Different questions to treat different topics can be part of the same exercise, that is why we assign the *topic* attribute to the question concept and not to the exercise. The answer focus, which is a chunk, consists of a *head* which contains the necessary information of the chunk to treat the topic. For instance, if we are working on the ergative, we distinguish the word of the chunk containing the mentioned linguistic phenomenon that is represented into the head tag. The rest of the words of the chunk and their linguistic analysis become part of the *notHead*. In addition, depending on the question type, the answer focus can be a blank in the stem; the *blank* attribute offers this option.

The representation of the question model also permits to change the order of the chunks in the sentence. This is very important as we will see in section 4. There, the expert teachers propose, for example, to change the order of the blank inside the generated question. This is frequent in languages which have not a very strict order for the phrases of a sentence. Therefore, we have defined *posQ* and *posS* attributes: posS represents the position of the chunk in the source sentence and posQ in the question. Moreover, the *change* attribute limits which chunks can change the order when setting the final question in the assessment application.

The head, which is the list of words where the topic in matter appears, is divided into three parts: the *answer*, the list of *distractors* and the list of *headComponents*. The answer is the only mandatory attribute and the other two take part depending on the question type. We define the answer as the minimum list of *words* where the topic to treat appears, the *topic info* and the *analysis*. The *headComponent* collects the specific information related to the *question type***.**

As regard distractors, we define a distractor as a list of words which are incorrect in that context. That is why they are always linked to an answer focus. The model do not only offer the list of words and their corresponding linguistic analysis, but it also offers the heuristics used for creating each distractor. For example, in section 4.2 we present the evaluation of multiple-choice questions about the conjugation of verbs in Basque language. The heuristics change, among others, person and number of the subject in the correct verb form in order to generate different distractors.

The type of questions for language learning created by other studies ([3], [6], [7], [9], [11], [14]) can be represented by our question model. However, we ought to specify that in the case of error correction in [4] what they consider distractors, i.e. underlined chunks, in our case, it is a different concept. In our model, the distractor should be just an incorrect choice of the answer focus. Thanks to the heuristic and the topic information given in the head, we are able to represent such information. In this way, the assessment application using the questions generated by ArikIturri is able to detect those chunks to be underlined.

Next section deals with an experiment carried out using the model we have presented.

4. Evaluation

For the manual evaluation of the questions, we have implemented a web-based post-editing environment for helping teachers to set the questions. Next, we will explain some features of the environment, the experiment and the results.

4.1. The Post-editing environment

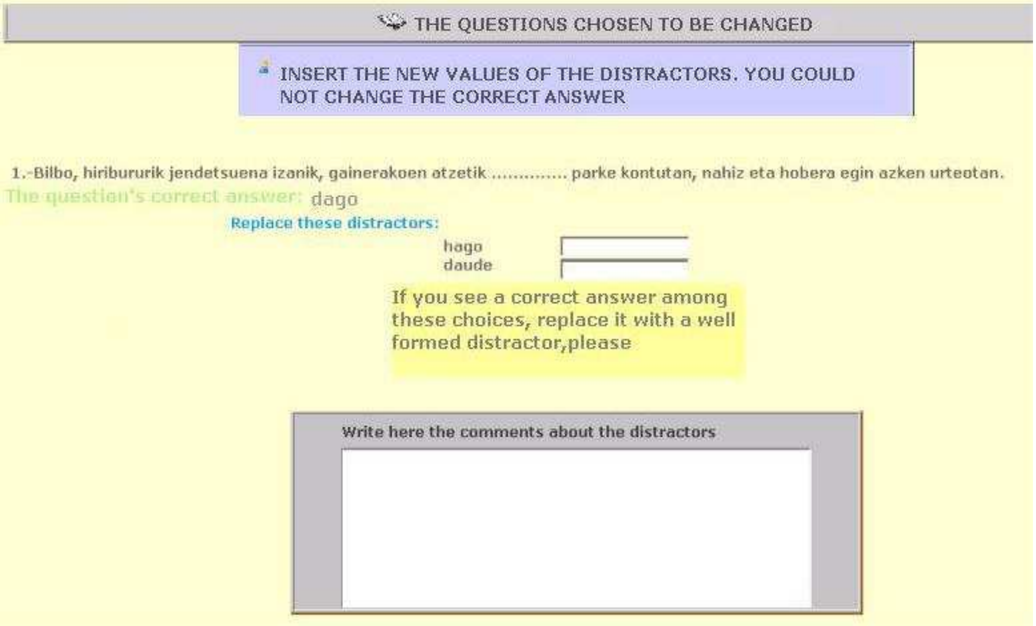
The post-editing environment requests ArikIturri to generate language questions. Those questions, which are represented by means of the question model, are imported to the environment's database. Post-editors, i.e. expert teachers, can modify some components of those automatically generated questions.

Thanks to our post-editing environment, teachers have different options regarding each generated question: to accept it on its own, to discard it if it is not an appropriate question or to modify it. As the rejection or modification of a question can involve a feedback process for improving ArikIturri, the environment offers the option to add comments related to the reasons for not accepting or modifying the question (see figure 2). Moreover, if the post-editor decides to modify a question, there are several possibilities. Firstly, the source sentence can be modified if, for instance, some misspellings appear, but, post-editors can never modify the correct answer. Secondly, when distractors are generated they can update them or add new ones if they consider that, among the options, there are more than one possible correct answer or other distractors are more appropriate. Taking into account the collected information from the post-editors, the system will be in a continuous progress. Deeper research in the way of providing feedback to ArikIturri will be done in the near future.

4.2. The experiment

In [1] we made some experiments in order to evaluate the correctness of the automatically generated questions. There, we used a computer assessment application for the evaluation. In that case, the application did not offer the option to explain the reasons for discarding or updating the questions. In this new experiment we use a post-editing environment especially designed for evaluating the quality of the generated questions. The environment gives to the human evaluator the opportunity to explain its actions. Concretely, they will explain their reasons to accept, discard or modify a stem or a distractor.

The experiment is focused on the multiple-choice question type, as the generation of distractors is a matter of high difficulty when setting language questions. This generation is even more difficult when the distractors are not taken from a databank, but they are automatically generated words. The topic of the questions that the generator has to select is the conjugation of verbs. This topic is a difficult task in Basque



THE QUESTIONS CHOSEN TO BE CHANGED

INSERT THE NEW VALUES OF THE DISTRACTORS. YOU COULD NOT CHANGE THE CORRECT ANSWER

1.-Bilbo, hiribururik jendetsuena izanik, gainerakoen atzetik .... parke kontutan, nahiz eta hobera egin azken urteotan.

The question's correct answer: dago

Replace these distractors:

hago
daude

If you see a correct answer among these choices, replace it with a well formed distractor,please

Write here the comments about the distractors

Figure 2 - The post-editing environment

language as, when making verb's conjugation, the verb form changes depending on the person and number of the ergative, absolutive and dative cases of the verb paradigm. For example, the verb *izan (to be)* has 279 different forms for present tense. The experiment has been done with 597 sentences selected out of 10079. These sentences correspond to 234 texts chosen for learning purposes of high language level students. The human resources were two expert teachers working for eight hours each. Taking as a basis this restriction we adjusted the number of selected sentences for the tasks of the experiment.

The main objective of the first task of the experiment is twofold. In the one hand, as the extraction of the candidate sentences is an automatic process, we want to identify which kind of changes post-editors propose for the candidate sentences when setting questions about specific topics. On the other hand, we intend to pick out the reasons for discarding or updating the automatically generated distractors of multiple-choice questions. The experiment is a first step in the way of improving the quality of the generation.

In addition, we have designed a second task of the experiment to request post-editors (expert teachers) to create distractors by hand in order to obtain their experience.

4.3. Results

Two expert language teachers of HABE, an institute of the Basque government for L2 and L1 Basque Language Teaching, used the post-editing environment. They evaluated the appropriateness of 597 candidate sentences (stems) of multiple-choice questions. The 8.87 % of the questions were discarded. However, it is important to remark that the experiment was made by means of two different kinds of tasks. In the first task both post-editors could consult the automatically generated distractors as well as the correct answer of the source text. In this task they discarded only the 2.55 % of the 392 stems. In the second task, one of the post-editors analyzed the quality of the stems without having the chance of consulting any possible distractor. In this case, he discarded the 20.97 % of the 205 automatically generated questions. We presume that the possibility of consulting the generated distractors had influence on the results. In our opinion, this information gave them a more restricted perspective of the topic to be analyzed.

Next, we will comment on the main *reasons for discarding the stems*. In the one hand, they founded some stems with more than one correct answer. They considered that it was easier to discard those sentences than to change their distractors. In the other hand, some of the sentences were difficult to understand. Sometimes, the post-editors needed a longer context of the sentence in order to understand the topic of the question. In other cases, the ellipsis of some phrases of the sentence made difficult the identification of the correct form to fill the blank. Finally, it is important to underlie that only one sentence was discarded because the blank of the question did not correspond to the selected topic, i.e. the verb.

In this paragraph, we are commenting on the main *reasons for updating the stems*. The post-editors cut sentences that they considered too long. They also made changes when they considered style aspects of the sentence or incorrectness respect to the standard definition of Basque grammar. This is an important aspect as the normalization process of Basque is currently in progress. The position of the blank of the question was also a reason to update the stem, for example, they proposed to change the position of the blank position if it was at the beginning of the question.

The quality of multiple-choice questions also depends on the quality of the generated distractors. In the experiment for evaluating the appropriateness of the generated distractors the results were quite good as only the 2.04 % of the generated questions were discarded as a consequence of the evaluation of the distractors. However, we have to say that among the rest of the questions, the 91.83 % were accepted and the 6.12 % were updated because of different reasons. For example, one of the main *reasons for discarding and updating the generated distractors* is that the generator gives as results some distractors that can be correct answers. Deeper research studies of these results, (the 6.12 % and the 2.04 %), will give us hints to improve the heuristics of the generator.

5. Conclusions and future Work

In the present work we have explained the question model of Arikiturri, an automatic question generator. Moreover, we have commented on the results of an experiment. This experiment is a first step on the improvement of the quality of the automatic generation of multiple-choice questions.

Two main characteristics of the question model we present in this paper are generality and flexibility. It is a general model because of its multilingualism feature: the model is independent from the language of the generated questions. Furthermore, it is also independent from the NLP tools used in the generation. Indeed, our model allows different types of questions to be represented and, in addition, different types of questions can be specified into the same exercise. Finally, because the model has been developed using XML, the importation and exportation processes are easy tasks.

At the beginning of our work we studied some standards such as the IMS Question & Test Interoperability (QTI). As explained in section 3.2 the question model represents some data related to the generation process, e.g. the heuristics used for creating the distractors. This kind of information is not explicitly defined in QTI; that is why we decided to define our own model. Nevertheless, an extension of IMS-QTI to express the mentioned data will be considered as the next step of our research work.

The fact that the distractors of a head could have been created with different heuristics is a favourable option in terms of quality. Teachers can consult the automatically generated distractors and give advices or opinions about each one in order to improve them. The results of the experiment show that this approach gives good results and that the quality of the generation of the distractors is a matter of further research. In the near future, we will analyze the experts' opinions in order to improve the heuristics of the question model. In addition, we will evaluate the results with learners.

Acknowledgements

We would like to thank Ana Arruarte, teacher and researcher of the University of the Basque Country for her suggestions when writing this paper. We also are grateful of the expert teachers, Jon Imanol Iturrioz and Julio Aparicio, as well as of HABE, their institution that has assigned good expert teachers in order to evaluate the quality of the questions.

References

[1] I. Aldabe, M. Lopez de Lacalle, M. Maritxalar, E. Martinez, L. Uria, "ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques", *Proceedings of the Eight International Conference on Intelligent Tutoring Systems (ITS'06)*, pages 584-594, Jhongli, Taiwan, 26-30 June 2006.

[2] I. Aldabe, M. Lopez de Lacalle, M. Maritxalar, E. Martinez, "The Question Model inside Arikiturri", *Proceedings of the ICALT Conference 2007. Incoming.*

[3] J.C. Brown, G.A. Frishkoff, M. Eskenazi, "Automatic Question Generation for Vocabulary Assessment", *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 819-826, Vancouver, October 2005.

[4] C. Chen, H. Liou, J.S. Chang, "FAST – An Automatic Generation System for Grammar Tests", *Proceeding of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1-4, Sydney, July 2006.

[5] E. Guzmán, E. Machuca, R. Conejo, P. Libbrecht, "LEACTIVEMATH Integrated Adaptive Assessment Tool", *The LEACTIVEMATH Consortium*, July 2005.

[6] A. Hoshino, H. Nakagawa, "A real-time multiple-choice question generation for language testing – a preliminary study", *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 17-20, Ann Arbor, June 2005.

[7] O. Kraif, G. Antoniadis, S. Echinard, M. Loiseau, T. Lebarbé, C. Ponton, "NLP Tools for CALL: the Simpler, the Better", *Proceedings of the InSTIL/ICALL2004 –* NLP and Speech Technologies in Advanced Language Learning Systems, Venice, 2004

[8] M. Larrañaga,, U. Rueda, J. A. Elorriaga, A. Arruarte. Acquisition of the Domain Structure from Document Indexes Using Heuristic Reasoning. In Proceedings of the Intelligent Tutoring Systems Conference, pages 175-186. Barceló (Brasil). 2004.

[9] C. Liu, C. Wang, Z. Gao, S. Huang, "Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items", *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 1-8, Ann Arbor, June 2005.

[10] M. Mavrikis, "MathQTI and the Serving Mathematics project", http://www.mathstore.ac.uk/articles/maths-caa-series/jan2005/.

[11] R. Mitkov, L.A. Ha, N. Karamis, "A computer-aided environment for generating multiple-choice test items", *Natural Language Engineering: Special Issue on using NLP for Educational Applications*, Cambridge University Press, 12 (2): 177-194. 2006.

[12] T. Murray Authoring Intelligent Tutoring Systems: an Analysis of the State of the Art. International Journal of Artificial Intelligence in Education, 10, 98-129. 1999.

[13] M.J. Pollock, C.D. Whittington, G.F. Doughty, "Evaluating the Costs and Benefits of Changing to CAA", *Proceedings of the Fourth International Computer Assisted Conference CAA*, 2000.

[14] E. Sumita, F. Sugaya, S. Yamamoto, "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*, pages 61-68, Ann Arbor, June 2005.

[15] Vereoustre, A. and McLean, A. "Reusing Educational Material for Teaching and Learning: Current Approaches and Directions". In Supplementary Proceedings of AIED2003, Aleven, V., Hoppe, U., Kay, J., Mizoguchi, R., Pain, H., Verdejo, F. and Yacef, K. (Eds.), pp. 621-630. 2003.