Euskal Herriko Unibertsitatea

eman ta zabal zazu

Informatika Fakultatea
Lengoaia eta Sistema Informatikoak Saila

Automatic Exercise Generation Based on Corpora and Natural Language Processing Techniques

Itziar Aldabe Arregik

Informatikan Doktore titulua eskuratzeko aurkezturiko
Tesi-Txostena

Donostia, 2011ko iraila

Euskal Herriko Unibertsitatea

eman ta zabal zazu

Informatika Fakultatea
Lengoaia eta Sistema Informatikoak Saila

Automatic Exercise Generation Based on Corpora and Natural Language Processing Techniques

Itziar Aldabe Arregik Montse Maritxalar Angladaren zuzendaritzapean egindako tesiaren txostena, Euskal Herriko Unibertsitatean Informatikan Doktore titulua eskuratzeko aurkeztua

Donostia, 2011ko iraila

Decide what to be and go be it
The Avett Brothers

Rafari

Acknowledgements — Esker onak

Montse, eskerrik asko tesian eskainitako laguntzarengatik; ikerkuntzan barneratzen laguntzeagatik. Primerako tesi-zuzendaria izatez gain, momentu kritikoetan gertu sentitu zaitut.

IXA taldeko kide eta kide ohi guztiei ere, eskerrik asko. Piramideko goialdean kokatzen den horrelako tesi bat ez litzateke bideragarria beste askok egindako lanarengatik ez balitz.

HIREKIN azpitaldeko partaide izanda, ezin ahaztu elkarrekin egindako bilera eta lan guztiak. Oso aberasgarria izan da zuekin lan egitea.

Mila esker ArikIturriren sortzaile diren Edurne Martinez eta Montseri, ikerketa-lerro honekin hasteagatik.

ArikIturriren ariketak sortzeko determinatzaileekin egindako lana Larraitzi eta Olatzi zor diet. Zuei esker erroredun ariketak sortzeko gai da ArikIturri.

I would like to mention Professor Ruslan Mitkov and the Research Group in Computational Linguistics from the University of Wolverhampton. My stay in Wolverhampton enabled me to start working with English language items. Thanks for the meetings and the ideas arisen in them as well as your help in the manual evaluation. It was a pleasure to meet you all. You made me feel at home.

Lana errazteko eskaini didazuen laguntza bereziki eskertu nahi dizuet bakar batzuei: Aitziber, aditzen patroiak eskuragarri jartzeagatik; Bertol, perpausen mugak markatzeko beharrezko guztiak modu errazean erakusteagatik;

Aitor, grafoekin laguntzeagatik; XArregi, galderen sorkuntzaren hasierako pausoak ematen laguntzeagatik; Ander, galderen sorkuntzarako zenbakiak markatzeagatik; eta Iñigo, galderen sorkuntzarekin laguntzeagatik.

Ebaluazioa, tesi lan honen parte garrantzitsu bat izan da. Eskuzko ebaluaziorako, taldeko zein kanpoko jende askoren laguntza izan dut eta beraien izenak aipatzea ezinbestekoa zait. Hizkuntzalariak: Larraitz, Maxux, Eli Izagirre eta Ainara; Informatikariak: Kike, Montse eta Koldo; Maddalen, Makulu eta Kalitarik aplikazioen sortzailea; HABEko Jon Imanol eta Julio; Wolverhampton unibertsitateko Kathryn eta Judith; Arantxa eta Daryl; Argentinako euskal etxeko Sabrina eta bere ikasleak; eta azkenik Ikastolen Elkarteko kideak. Azken hauen laguntza bereziki aipatu nahiko nuke. Ikastoletako irakasle zein ikasleen bidez eta Josune Gereka eta Ramon Gorrotxategiren lanari esker, izugarrizko ebaluazioa burutzeko aukera izan dut.

Tesiaren zuzenketan lagundu didazuen guztiei (German, Nerea, Gorka, Oier eta Ruben): thanks!

Bulegoa edo bazkal orduak konpartitutako guztiei ere nire eskerrik beroena.

Quiero nombrar también a los aitas. Gracias a vosotros he llegado hasta aquí. Rafa, Gloria, agian zuek ez zarete konturatu, baina zuek erakutsi didazue ikerketa hautabide bat badela. Ilobei... jenialak zarete!

Lagunak: zuek ere jenialak zarete!

Azkenik Oier aipatu nahiko nuke. Ezin nahikoa eskertu ondoan izateagatik eta eman didazun laguntza guztiagatik (tesian, egunerokoan...). Eskerrik asko bihotz-bihotzez eta muxu bat.

Acronyms and abbreviations

ArikIturri:	Ariketen Iturria. Iturria means “fountain” and Ariketen means “of exercises”
AWL:	Academic Word List
BNC:	British National Corpus
CAE:	Certificate in Advance English
CEFR:	Common European Framework of Reference for Languages: Learning, Teaching, Assesmment
CG:	Constraint Grammar
CTT:	Classical Test Theory
DTD:	Document Type Definition
EFL:	English as a Foreign Language
EGA:	Euskararen Gaitasun Agiria
EPEC:	Euskararen Prozemanendurako Erreferentzia Corpusa
FBQ:	Fill-in-the-Blank Question
GSL:	General Service List
HABE:	Institute for the Teaching of Basque and Basque Language Literacy to Adults.
HEOK:	Helduen Euskalduntzearen Oinarrizko Kurrikulua
ICT:	Information and Communication Technologies
IMS:	IMS Global Learning Consortium
LKB:	Lexical Knowledge Base
LMS:	Learning Management System
LSA:	Latent Semantic Analysis
MCQ:	Multiple Choice Question
MCR:	Multilingual Central Repository
NLP:	Natural Language Processing
NuERCB:	Numerical Entity Recogniser and Classifier for Basque

OOP:	Object-Oriented Paradigm
OSE:	Obligatory Secondary Education
PoS:	Part-of-Speech
QG:	Question Generation
QTI:	Question and Test Interoperability
SVD:	Singular Value Decomposition
UML:	Unified Modeling Language
VSM:	Vector Space Model
XFST:	Xerox Finite-State Tool
XML:	eXtensible Markup Language
XSD:	W3C XML Schema
ZT corpus:	Zientzia eta Teknologiaren Corpusa

Contents

CHAPTER I

Introduction

I.1 Motivation and context

Nowadays, the introduction of information and communication technologies (ICT) into educational areas is a reality. In fact, ICT competence is one part of the National Curriculum in the United Kingdom, the Australian Curriculum and the Basque Curriculum. In 2008, even UNESCO published ICT competency standards for teachers. Various official institutions are investing money in introducing technology into classrooms. Evidently, the use of ICT is not exclusive to classrooms. For instance, ICTs are widely used in distance learning scenarios. In fact, nowadays, no understanding of distance learning exists which does not involve ICTs. Two illustrative examples from the Basque Country are two learning management systems (LMSs): Eskola 2.0[1] and Ikasys.[2]

In 2009, the Basque government started up the Eskola 2.0 programme with the principal aims of digitalising the classrooms in primary schools, training teachers in how to use free software and new technologies and, finally, producing multimedia content. Although it is a complete programme, one of its most famous actions was to provide one laptop per child in primary schools. However, although it is important to supply students with new media in order to encourage them in their learning process, it is indispensable

[1] http://www.eskola20.euskadi.net/

[2] http://www.ikasys.net

to provide them with high-quality content that can be used through these new technologies.

Some years before, in 2000, the Ikastolen Elkartea[3] detected such a need and started developing the Ikasys project to produce resources for primary and secondary education. Ikasys aimed to help students in a personalised way to achieve certain competencies and to respond to pedagogic needs. With this purpose, applications and content were developed, leading to not only a tool (the hardware and software) to be used within the classroom but also the content to improve the performance of students.

One institution which has experience of distance learning is HABE, the Institute for the Teaching of Basque and Basque Language Literacy to Adults. Since its creation in 1981, HABE has worked continuously in the areas of: (i) the design and execution of the curricula for the learning or teaching adults to read and write in Basque; (ii) the production of didactic material; (iii) teacher training; and (iv) offering students the necessary tools to learn and improve their knowledge of Basque. These objectives have led to a huge amount of material which could easily be distributed to a high number of users. For instance, HABE provides different learning-oriented material through Ikasbil.[4]

All of these examples are undoubtedly very useful resources for learners, but the tools which are produced usually contain previously established domains and content. As regards the exercises, these types of system can offer different types of exercise based on pedagogic criteria that can be adapted to meet learners' needs by means of artificial intelligence techniques. However, this pool of exercises is usually static, that is to say, the collection of items is always the same. **Our challenge is to turn this static knowledge into dynamic knowledge.**

As regards the creation of exercises, there are an increasing number of tools available to facilitate this task (Boyle *et al.*, 2004; Conejo *et al.*, 2004; Kerejeta *et al.*, 2005). However, teachers or experts usually have to insert the items manually.[5] The main objective of this dissertation is to demonstrate that there is also the option of automatically generating the exercises by means of natural language processing (NLP) techniques and corpora. As will be expounded in following chapters, the main purpose of this disserta-

[3] The Confederation of Basque private schools.

[4] http://www.ikasbil.net/jetspeed/

[5] In this dissertation, the terms "item" and "question" are used to refer to the output of ArikIturri.

tion is to offer those exercises to teachers and experts in order to make their task easier. We will prove that ArikIturri can help when creating didactic resources. For this reason, different techniques will be used in the generation process, including verb subcategorisation, morphological and syntactic analysis and semantics. Thus, based on these techniques, the automatic generation of exercises will be carried out in different domains (language learning and science) for different languages (Basque and English). **Therefore, the outcome of this PhD is a system called ArikIturri which creates exercises to be used in different domains**.

The heterogeneity of the output of the system is not an obstacle which will prevent the automatically generated content from being shared with LMSs. In fact, the purpose of our work is to offer the items in a structured way. For this reason, **ArikIturri is based on a question model in which items and information relating to their generation process are represented**.

I.2 Question, stem and distractor concepts

This section explains three essential concepts of this dissertation: **questions**, **stems** and **distractors**. Example I.2.1 presents a sample from a real test in order to illustrate these concepts.

Example I.2.1 (Sample of a real test — Basque — Science domain)

...1... hori behar bezala ez kanporatzeagatik, gerta daiteke birikek duten baino leku gehiago behar izatea aireak, eta presio horrek biriketako albeoloei eztanda eginaraztea. Albeoloak lehertzean sortzen den odoljarioak ...2... ditu, bai urpekariak, bai saguzarrak.[6]

1 a. Aire b. Haize c. Oxigeno d. Gas
2 a. gaixotzen b. mareatzen c. akabatzen d. desorientatzen

Although these concepts will be presented in depth in the following sections, we consider it necessary to explain them in order to assist with the

[6] If it is not properly expelled, ...1... may need more space than that which is provided by the lungs and so the pressure may cause the alveoli in the lungs to explode. The haemorrhage produced when the alveoli burst ...2... scuba divers and bats.
1 a. air b. wind c. oxygen d. gas
2 a. gets ill b. faints c. kills d. disorientates

reading of this chapter. Broadly speaking, a **question** or item is composed of a stem that requires an answer (key). The **stem** is the part of the item that presents the item as a problem to be solved, a question or an incomplete statement. Thus, the stem can be a declarative or interrogative statement. It can also be an incomplete sentence (containing a blank), and the correct answer to the stem is the key of the question. In addition, depending on the type of question, an item can also be composed of a list of distractors, a **distractor** being an incorrect choice among multiple-choice answers on a test.

Example I.2.1, which corresponds to a test which deals with scientific vocabulary, contains two multiple-choice questions (MCQs) and each MCQ consists of a stem and a set of options. In the given example, each stem is an affirmative statement accompanied by a blank. Each blank has different options; the correct answer is the key and the incorrect answers are the distractors.

There are various NLP-based approaches which have proven that the automatic generation of items is viable and the most commonly studied type of question is the MCQ. Some are focused on testing the use of grammatical knowledge (Hoshino and Nakagawa, 2007). Others work with semantics in order to test the student's knowledge of English (Pino *et al.*, 2008) or specific domains (Mitkov *et al.*, 2009).

I.3 Main objectives

This research work uses a multidisciplinary approach, through which two main objectives were defined. On the one hand, we want to study and exploit different NLP tools and resources in order to generate questions in an automatic way. On the other hand, we want to obtain pedagogically useful questions.

Thus, one of the objectives of this dissertation is to design and implement a system, called **ArikIturri**, which generates questions automatically.[7] Our purpose is to offer a broad and multilingual system and to prove these two features by means of different experiments. More specifically, the aim is to define experiments focusing on different scenarios: Basque language learning, English language learning and learning in the science domain. Therefore, the

[7] These questions will be part of an exercise.

aim is to use raw texts as the input for ArikIturri and to analyse them by means of NLP tools.

As a consequence, we foresee that the outcome of the system will be not only a set of questions but also information relating to the generation process. In order to represent this information in its entirety, we consider it to be necessary to define and implement a question model. In this way, it is possible to represent both input and output in a structured way. In addition, describing the questions based on a model facilitates the process of exporting them. This is an interesting feature as ArikIturri aims to be independent of any assessment application.

In addition, we consider it to be necessary to study and apply different methodologies in order to generate questions and their components. In fact, one of the strong points of our research is related to the automatic generation of distractors. The purpose of this is to perform an in-depth analysis of the available resources and to suggest non-supervised methods. However, we predict that the manual supervision of the resources will still be necessary. This is why it is important to analyse the available NLP tools and resources.

The other main objective of our work is to generate useful items, that is to say, questions that make students doubt or think about the topic in question. For this purpose, it is necessary to investigate different automatic methods in order to generate several question types, stems, keys and distractors.

However, the process of analysing the usefulness of the items is even more important. Therefore, various methods and theories will be applied in order to study the questions and their components. From our point of view, experts' opinions and students' responses will play an important role in the evaluation of the system. Moreover, this evaluation can be carried out with different purposes: to study the correctness of the items and to judge their quality. In both cases, the results can give us hints as to how to improve ArikIturri. In the end, these experiments will provide us with a way to judge the viability of offering ArikIturri as a tool to help experts and teachers in the generation of exercises.

I.4 Contributions

The main outcome of this dissertation is a system called ArikIturri. ArikIturri is a system which not only automatically generates items but also provides items of a certain quality. More specifically, we have proven the viability of this system to create different types of question: error correction, fill-in-the-blank questions (FBQ), word formation, MCQ and short answer questions. Moreover, **we have designed, implemented and analysed items to work in different scenarios**: Basque language learning, English language learning and learning in the science domain. The results of the experiments verified that ArikIturri can be offered as a tool for assisting with the generation of tests.

We have developed a system which is modular, multilingual and independent of any application. Thanks to its modularity, the adaptation of the system as well as the addition of new features is easy to carry out. The process of updating the system in order to generate questions in Basque and English proved this feature. In addition to designing a modular system which facilitates, among other things, the reusability and portability of the system, **we have also specified a question model** to represent the items generated by ArikIturri. Representing the items in a structured way makes the items accessible to the scientific community. In fact, in addition to our own model, the obtained items can also be represented by means of the IMS Question and Test Interoperability (QTI) standard (IMS Global Learning Consortium, accessed 2010). For this reason, we have proposed an extension point of QTI to insert some new information. This is due to the fact that our model comprises not only questions but also information relating to the process of their generation. All of the information offered by the question model accomplishes one purpose. In addition, the information relating to the generation process allows experts to study the generation process and subsequently to provide the system with feedback. **The two main characteristics of our question model are generality and flexibility**. It is a general model because of its independence from the language of the questions as well as from the tools used for their generation. Indeed, our model allows different types of question to be represented and, in addition, several types of question can be specified in the same exercise. Finally, because the model has been developed using extensible markup language (XML),[8] the import and export processes are easy tasks.

The two main resources used by ArikIturri to produce items are NLP tools and corpora. In fact, the entire generation process is based on these two resources. NLP tools analyse language automatically, which is why an error rate is always expected. The corpora in question are collections

[8] http://www.w3.org/XML/

of texts which have been selected with one particular purpose. Therefore, they cannot contain all the available knowledge of a language. Based on both premises, we have measured the influence that NLP tools and corpora can have on the generation process. The results of this analysis have shown that it is not possible to deal with some topics nor to define some heuristics.

Apart from the analysis of the resources, **we have also focused on the evaluation of the items**. On the one hand, we have measured the correctness of the items. This evaluation was carried out based on experts' opinions within a post-editing environment. On the other hand, we have judged the quality of the generated items. Their quality was judged by conducting various experiments with students. The results obtained show that the system is able to create useful exercises for testing the knowledge of students.

The use of **grammatical information** in the generation of items has proven that the system is able to generate correct questions to be used in a Basque language learning scenario in a completely automatic way. In this way, we have studied different methods for the automatic identification of topics as well as well as the automatic generation of different components of the questions. First of all, we examined how to transform declarative source sentences into interrogative ones in order to use them as the stems of the items. Second, we generated tests in order to deal with cases of declension, determiners and verb forms based on grammatical information. In order to analyse the correctness of these questions, the evaluation of these tasks was carried out by experts. Finally, the first steps have been taken towards the creation of heuristics based on automatically extracted grammatical information.

We have also worked with the **semantics** of the Basque and English languages in two different domains: language learning (English) and science (Basque). More specifically, we focused on the automatic generation of incorrect options (distractors) for MCQs. For this reason, we explored different methods of measuring relatedness between words and analysed the results both qualitatively and quantitatively. The evaluation of this task took into account the opinions of experts as well as students' answers. Thus, the generated items have been tested in a real scenario. The results show that we are able to offer experts a helpful tool with which to create didactic material.

I.5 Structure

The motivation, objectives and contributions of this dissertation have now been presented. In the following, the structure of the rest of this thesis is briefly explained.

- **Chapter II — ArikIturri**
  This chapter introduces ArikIturri, the system developed for the automatic generation of different types of question. It first describes the main features of the system. After that, the chapter presents the types of question implemented during the development of the system, as well as the necessary explanations regarding the topic concept. Finally, the architecture is expounded. The multilinguality and modularity of the system are justified and the main modules are described.

- **Chapter III — Question model**
  This chapter is devoted to the explanation of all of the concepts relating to the question model. For this purpose, the reasons for which such a model was defined are presented together with two of its main characteristics: its generality and flexibility. In addition, once the reasons for first defining our own model and then providing QTI with an extension point are determined, the main components of the structure as well as real instances of different types of question are displayed.

- **Chapter IV — Data analysis**
  In this chapter we present an analysis of the resources used by ArikIturri. In order to better understand the experiments presented in chapters V and VI, the main features of these resources are explained. In addition, the influence they can have on the generation process is also studied. Finally, the methods used to evaluate the automatically generated items are proposed. Moreover, the role played by experts in the evaluation as well as the applications provided for it are clearly expounded.

- **Chapter V — Using grammar when generating test items**
  This chapter presents the experiments in which grammatical information is used. Three main experiments are presented: (i) the generation of interrogative statements to be part of the stem of an item; (ii) the use of manually defined heuristics to deal with declension, determiners and

verbs in a Basque language learning scenario; and (iii) the first steps towards the creation of heuristics based on automatically extracted grammatical information. All of these experiments are evaluated with the help of experts in the field.

- **Chapter VI — Using semantics when generating test items**
  The experiments in which ArikIturri applies semantic information in order to generate items are grouped together in this chapter. First of all, the multilinguality of the system is proven by means of the generation of items designed to deal with English vocabulary (verbs). Moreover, the first heuristics which make use of semantic information are defined and evaluated based on experts' experience. Second, semantic information is also applied in the science domain in order to work with the vocabulary taught in Basque at secondary schools. This last set of experiments simulates the entire testing process by obtaining results from students.

- **Chapter VII — Conclusions and future work**
  The last chapter of this dissertation summarises the main conclusions and future work relating to the improvement of ArikIturri and the opening of new lines of research.

- **Appendices**
  The different appendices group together all of the additional information which is helpful for a deeper reading but that is not indispensable in order to understand this thesis. The appendices comprise: (i) the particularities of the Basque language as regards linguistic phenomena; (ii) the complete specification of the question model (this specification is presented in an XML schema); (iii) the details of the basic curricula for the process of learning the Basque language for adults in relation to our experiments; and (iv) the complete specification of the heuristics as regards the generation of tests designed to deal with determiners. This specification is also provided in an XML schema.

I.6 Publications

All the publications closely related to this thesis (subsection I.6.1) are following listed. In addition, author's publications that are related to NLP (subsection I.6.2) are also presented.

I.6.1 Publications related to this dissertation

- Aldabe I., Maritxalar M., Soraluze A. Question Qeneration Based on Numerical Entities in Basque. *In Proceedings of AAAI Symposium on Question Generation.* to appear, 2011. (**Chapter V**)

- Aldabe I., Maritxalar M. Automatic Distractor Generation for Domain Specific Texts. *ICETAL: Proceedings of the 7th International Conference on NLP, IceTAL 2010.* pp. 27-38. Reykjavik, Iceland, 2010 (**Chapter VI**)

- Aldabe I., Maritxalar M., Mitkov R. A Study on the Automatic Selection of Candidate Sentences and Distractors. *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009).* pp. 656-658. Brighton (UK), 2009. (**Chapter VI**)

- Aldabe I., Lopez de Lacalle M., Maritxalar M., Martínez E. The Question Model inside ArikIturri. *Proceedings of the 7th IEEE International Conference on Advance Learning Technologies (ICALT 2007).* pp. 758-759. Niigata (Japan), 2007. (**Chapter III**)

- Aldabe I., Maritxalar M., Martínez E. Evaluating and Improving Distractor-Generating Heuristics. *Proceedings of the Workshop on NLP for Educational Resources.* In conjuction with RANLP07. Ezeiza, N., Maritxalar, M. and Schulze M. (Eds). pp. 7-13. 2007 (**Chapter V**)

- Aldabe I., Lopez de Lacalle M., Maritxalar M. Automatic acquisition of didactic resources: generating test-based questions. *Proceeding of SINTICE 07 (Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación).* pp. 105-111. Zaragoza (Spain), 2007. (**Chapters III and V**)

- Aldabe I., Lopez de Lacalle M., Maritxalar M., Martinez E., Uria L. ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. *In Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS'06).* pp. 584-594. Jhongli (Taiwan), 2006. (**Chapters II and V**)

I.6.2 Publications related to the field

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. Hizkuntzaren Tratamendu Automatikoa Euskararen Irakaskuntzan. *BAT Soziolinguistika aldizkaria.* number 66, pp. 61-69. 2008.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L., Amoros L. Learner and Error Corpora Based Computational Systems. *In Corpora and ICT in Language Studies: PALC 2005.* J. Walinski, K. Kredens & S. Gozdz-Roszkowski (eds.), Peter Lang. Vol. 13. 2007.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. Basque error corpora: a framework to classify and store it. *In the Proceedings of the 4th Corpus Linguistic Conference online archive: http://www.corpus.bham.ac.uk/corplingproceedings07/.* 2007.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. The Use of NLP tools for Basque in a multiple user CALL environment and its feedback. *in Proceedings of TAL & ALAO workshop. In Proceedings of the 13th Conference Sur Le Traitement Automatique des Langues Naturelles.* Volume 2. p.: 815-824; Leuven, 2006.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L. Propuesta de una clasificación general y dinámica para la definición de errores. *Revista de Psicodidáctica, EHU.* Vol 10, N 2, p. 47-60 (ISSN: 1136-1034). 2005

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L., Amoros L. Leire Amoros IRAKAZI: a web-based system to assess the learning process of Basque language learners. *EuroCALL.* Cracovia, Polonia. 2005.

- Agirre E., Aldabe I., Lersundi M., Martinez D., Pociello E., Uria L. The Basque lexical-sample task. *In Proceedings of the 3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL).* pp. 1-4. Barcelona, Spain

CHAPTER II

ArikIturri: The origin of questions

In this chapter, we introduce ArikIturri, a system developed for the automatic generation of different types of question. One of the aims of ArikIturri is to generate items that could form part of real scenarios; this is why their creation is based on topics that are part of the curriculum. In order to create a basic understanding of ArikIturri, this chapter focuses on the explanation of the main features of the system as well as on the presentation of the main modules which comprise its architecture.

II.1 Introduction

Figure II.1 illustrates the main idea behind the ArikIturri system. ArikIturri is an contraction of "Ariketen Iturria," where Iturria means "*fountain*" and Ariketen means "*of exercises*." Therefore, the name refers to a system which has taken as its basis the work previously developed by Martinez (2005) and which is able to automatically generate tests from texts, to be included in testing tasks.

In the literature, we can find different pedagogical approaches to defining the learning process. Although there are other systems and hierarchies, Bloom's taxonomy (Bloom, 1956) is easily understood and is one of the most widely applied. In the 1950s, Bloom defined three domains of educational activity: cognitive (knowledge), affective (attitude) and psychomotor (skills). Thus, after a training session, the student should have acquired new skills,

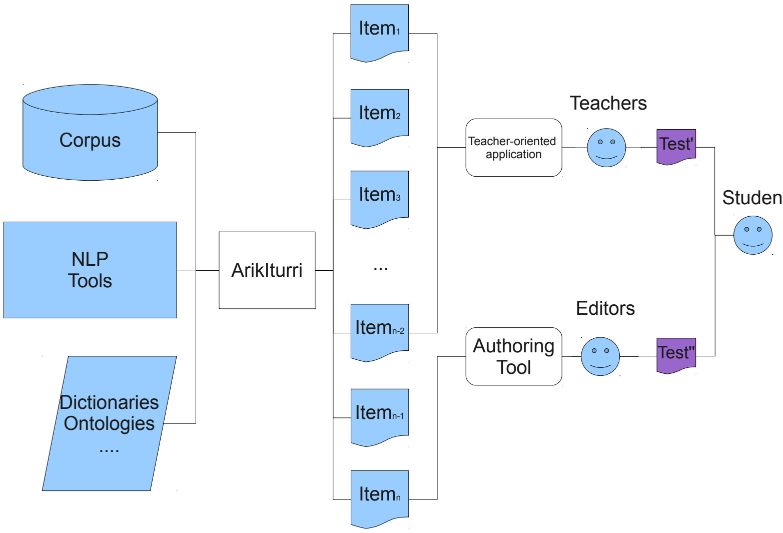


Figure II.1: ArikIturri

knowledge or attitudes. These three main domains can also be divided into subdivisions, from the simplest behaviour to the most complex.

In our approach, we focus on the cognitive domain that involves knowledge and the development of intellectual skills.[1] This includes the recall or recognition of specific facts, procedural patterns and concepts that serve in the development of intellectual abilities and skills. Therefore, our system aims to be a source of items with which to create a bank of useful exercises to be used within the cognitive domain. As the creation of tests is a difficult task even for human generators, a system which produces a draft of a test can be seen as a helpful tool for experts and teachers (Coniam, 1997).

In order to generate good-quality items, ArikIturri makes use of two kinds of language resource: NLP tools and linguistic information. These resources are used in different steps of the process and, as the architecture of the system is modular, they are clearly separated from the programs. In addition, ArikIturri is a system with an open architecture that allows the integration of new tools and resources. In fact, the two main characteristics of the system

[1] Bloom identified six levels within the cognitive domain, from the simplest to the most complex: knowledge, understanding, application, analysis, synthesis and evaluation.

are multilinguality and modularity (cf., section II.4). Apart from NLP tools and linguistic information, in this dissertation, the main source used to create the items is input texts which are collected in different types of corpus.

Items & item banks

As previously mentioned, ArikIturri is a tool which was defined in order to create different types of item to be part of an item bank. It is important to note that, in this dissertation, the terms item and question are used interchangeably, even if not all items contain interrogative statements. The terms item (Lee and Seneff, 2007) and question (Pino *et al.*, 2008) are used within the research community and although we mainly use the term question (Aldabe *et al.*, 2007b), the term item can be seen as a more general term.

In fact, as Vale (2006) states, the term item covers a variety of concepts, as tests are not always collections of questions, but problems to solve or even assertions to evaluate. Vale (2006) also suggests that an item, for the purpose of item banking, must include a stimulus. This stimulus can be a simple question, a question followed by several alternative answers or part of a larger structure consisting of other stimuli (Vale, 2006). In addition, the definition extracted from Vale (2006) is a working definition which is useful for the construction of item banks and appropriate from our point of view because one of the aims of this dissertation is to offer items and the information relating to them in a structured way.

As regards one possible representation of items and their information, the IMS Global Learning Consortium[2] has defined a specification to represent items and tests as object models. It is important to note that the IMS specification represents a standard of information exchange but that it is not a standard for item banking. Thus, its goal is to provide a representation in which items and tests can be exchanged among users working with a wide variety of applications. This is important for us because the portability of the generated questions is one of the aims of this work. Within IMS, items consist of two classes of elements: materials and interactions. Materials are anything that is presented to the examinee (the stimulus in Vale (2006)). An interaction is the method by which the candidate selects or creates a response. These data are represented by the IMS QTI standard (IMS Global

[2] http://www.imsglobal.org/

Learning Consortium, accessed 2010) which is expressed in XML.

In addition, an item bank is more than a collection of items or questions, as the items usually have different properties which lead to the specification of the information relating to their administration and scoring and QTI, for instance, foresees the representation of such information. However, ArikIturri does not aim to generate a complete item bank, nor to provide items with all of the properties that they can contain within the bank. Therefore, the system tries to generate items which will constitute the starting point of such a bank. In contrast, the output of the system offers additional information relating to the generation process which can be useful for experts and which is not specified in QTI. As QTI offers a general standard, there is no option to represent this information explicitly by means of such a standard. This non-representable information is related to the generation process and it is stored with one purpose. ArikIturri is based on experts' knowledge as well as linguistic information in order to generate items. As one of its aims is the representation of this information, in order to do so, we first defined our own question model (cf., section III.3.1) which encompasses all of the information obtained from the generation process. However, we also propose the aforementioned model as an extension of the QTI standard (cf., section III.3.3) in order to offer it to the scientific community.

Apart from representing the information relating to the generation process, we also consider it necessary to represent explicitly the topic of each item. For us, the topic is the concept that students have to work with and is part of their curriculum. From an item banking point of view, this concept can be seen as the stimulus of the item. From a more pedagogical point of view, a topic is a concept that students work with during their learning process.

In conclusion, it is necessary to note that all of the information which cannot be represented explicitly in QTI is stored with certain purposes. For instance, information relating to the topic is useful in order to classify the item into an item bank as well as to easily create tests to deal with a concrete topic. The aim of storing heuristic information relates to experts and item bankers. This information is very useful for understanding the automatic generation of items. In addition, once experts are able to interpret the heuristics, the system can receive their feedback on the generation process.

Assessment & evaluation

Producing items of a certain quality is as important as generating items automatically. Thus, this dissertation not only proposes methods for producing items, but also various approaches for measuring their correctness. Although the final purpose of such items should be to assess students in real scenarios, in this dissertation, we focus on the evaluation of ArikIturri. In order to avoid any misunderstanding, we consider it necessary to distinguish between the concepts of assessment and evaluation.

Assessment measures the performance of learners with regard to a set of competencies such as knowledge, skills and attitudes. In addition, two of the main characteristics of a test designed to assess students are reliability and validity. Reliability is obtained when the same test is evaluated with the same group of students in different periods and the results obtained are the same. Validity is computed in order to ensure that the test measures what it is intended to measure.

Evaluation is the process of determining the value of the items in order to accept, modify or reject them. This evaluation provides us with hints to improve the automatic generation process, for instance, by modifying the heuristics that are used in the generation of distractors (cf., chapters V and VI). The evaluation of the system is carried out in two different ways: on the one hand, evaluating the questions with experts (qualitative analysis) and, on the other hand, giving the revised items to students in order to measure their quality (quantitative analysis).

Therefore, this work does not prove the reliability and validity of the tests explicitly, but the quality of the system. However, in the end, good-quality items could lead to the creation of tests with which to assess students in a learning scenario.

Portability

Thanks to the structured representation of the items, integrating them into a learning scenario is an easy task. This is something to take into account, as ArikIturri is independent of any application that uses its items (see Figure II.1). In this way, different applications, with distinct purposes, can use the items generated by the system. The range of applications goes from authoring tools to teacher-oriented applications; even some student-oriented applications could make use of the items generated by the system.

The method of importing the items will be explained in more detail in chapter III, but we shall now briefly explain an example of the generation process. An expert user of an authoring tool determines the type of items to be generated, as well as the topic to be addressed in these questions.[3] Once the request has been made, the questions generated automatically by ArikIturri must be imported to the application. The question model (cf., chapter III) represents not only the items, but also the corresponding topical information as well as the heuristics used for their generation. Thus, depending on the application, this additional information is also provided to the user.

II.2 Types of question

As previously mentioned, Bloom identified six levels within the cognitive domain. Some types of item are appropriate for the assessment of knowledge goals, while other types of item are better for assessing application, analysis, synthesis or evaluation goals. Students' knowledge level can be tested with items such as true/false questions and matching pairs, and the rest of the levels with MCQs, short answer and essays questions.

Although there are various types of test item that are commonly used in testing environments, the systems which are based on the automatic generation of items are generally more specific. Most of the works that use NLP tools or linguistic information deal with just one type of item, the most commonly used being MCQs[4] (Coniam, 1997; Liu *et al.*, 2005; Mitkov *et al.*, 2009; Sumita *et al.*, 2005; Smith *et al.*, 2009). In addition, some authors such as Pino *et al.* (2008) offer not only MCQs but also FBQs for which students have to produce a word as the answer.

Our system is able to generate more than one type of item. As will be proven in the following chapters, ArikIturri is able to produce **FBQ**, **word formation**, **MCQ**, **error correction** and **short answer questions**. Each type of question is explained together with an example. The examples of FBQs, word formation, MCQ and error correction questions are based on the following source sentence: Sintomak honako hauek dira: aldarte txarra, estresa eta antsietatea. (The symptoms are: bad-mood, stress and anxiety.). The input sentence for the example of short answer questions is Gabonetan

[3] It is possible to set more than one topic per request.

[4] Some authors refer to this type of question as FBQs and others as cloze items or exercises.

karroza-desfile ikusgarria egiten dute (At Christmas, an amazing float parade is held).

Fill-in-the-blank

Generally speaking, FBQs require students to complete a statement by supplying a brief response. In some cases, an FBQ can be a question which students have to answer with a brief response. One of the advantages of this type of item is that the students have to provide the correct answer instead of just recognising it. Example II.2.1 presents an automatically generated FBQ in which the topic is the conjugation of the verb.

Example II.2.1 (FBQ example)

Sintomak honako hauek .... : aldarte txarra, estresa eta antsietatea
(The symptoms ....: bad-mood, stress and anxiety.)

In our experiments, the FBQ consisted of a stem that was an incomplete statement, meaning that the item always had a sentence with at least one blank to be completed. Thus, from a computational point of view, FBQs are focused on the correct marking of the blanks. When deciding on what should be blank in the item, the system itself chooses which units to remove from the text: the unit can be a single word or a phrase. In addition, the system can construct questions with more than one blank and, depending on the exercise and the topic, each blank is filled with one or more words. Our experiments as regards FBQ (cf., section V.3) were focused on the correct use of the declension cases and verbs in the Basque language. Each item comprised one blank.

Word formation

Word formation items consist of a sentence with a blank and a word the form of which must be changed in order to fit it into the gap. Depending on the topic of the item, the word formation task can vary. For instance, in language exams such as the Certificate in Advanced English (CAE), word formation items are used to test the student's knowledge of word families and their formation. For this reason, these exams provide the root of the word family and, based on the given context, students look for clues which will tell them what kind of word (adjective, noun, verb, adverb) is missing.

Our approach is focused on the correct use of declension cases and verb tense and persons, so, the aim of the word formation items is different. Example II.2.2 exemplifies this difference.

Example II.2.2 (Word formation example)

Sintomak honako hauek .... (izan) : aldarte txarra, estresa eta antsietatea
(The symptoms .... (to be): bad-mood, stress and anxiety.)

Students have to find the correct tense and subject of the verb izan (to be). Thus, the information offered in brackets does not correspond to the root of a specific word family, but to the root of a word category. The word to be changed is the lemma of the answer that originally appeared in the sentence and to obtain it, the system uses a lemmatiser.

Multiple-choice questions

Although this type of item does not require a creative process, making a choice requires high-level thinking.[5] MCQs consist of a stem and a set of options. The stem is the first part of the item and presents the item as a problem to be solved, a question or an incomplete statement. The options are the possible answers that the students can choose from, with the correct answer (the *key*) and the incorrect answers (*distractors*).[6] In our approach, only one answer can be correct. Example II.2.3 presents a multiple-choice item.

Example II.2.3 (Multiple-choice example)

Sintomak honako hauek .... : aldarte txarra, estresa eta antsietatea
(The symptoms .... : bad-mood, stress and anxiety.)
*a) dira (are) (*key*)*
*b) da (is) (*distractor*)*
c) daude (are[7]*) (*distractor*)*

[5] From a computational point of view, this type of item is the most complex to generate, as the distractor generation task is carried out automatically.

[6] Some authors refer to them as *distracters*.

[7] The Basque verbs egon and izan correspond to the English verb to be.

When generating multiple-choice items, apart from automatically marking the blanks, the system also generates distractors. From a computational point of view, this is one of the most difficult tasks of the entire generation process, which varies depending on the topic. Therefore, ArikIturri makes use of various pieces of linguistic information and tools depending on the topic. This set of information is the basis of the heuristics, and each applied heuristic is represented together with each distractor within the question model (see chapter III).

The MCQs were generated in different scenarios and all of the MCQs were composed of stems which were declarative statements and contained at least one blank (cf., sections V.3.1, V.3.2, VI.3 and VI.4). However, the system is also able to generate stems which are interrogative statements (cf., section V.2).[8]

Error correction questions

Error correction items consist of a sentence with at least one error that students have to correct. The error, which can be marked or unmarked, is a distractor which is generated automatically by the system. Example II.2.4 shows an error correction question where the error is unmarked while example II.2.5 shows an item with a marked error.

Example II.2.4 (Error correction example — Unmarked)

Sintomak honako hauek da : aldarte txarra, estresa eta antsietatea
(The symptoms is: bad-mood, stress and anxiety.)

Example II.2.5 (Error correction example — Marked)

Sintomak honako hauek da : aldarte txarra, estresa eta antsietatea
(The symptoms is: bad-mood, stress and anxiety.)

This type of items was generated in order to deal with the grammar of the Basque language (cf., section V.3.1).

[8] In fact, the experiments which focused on the automatic generation of question statements have been designed to offer different stem types.

Short answer questions

Short answer items require students to respond to a question by generating a brief text or response. This item format requires that students not only provide the answer but also express their ideas. Thus, this type of item is usually used to assess high-level thinking within the cognitive domain.

We have distinguished two groups of short answer questions which are created by ArikIturri. Both comprise interrogative statements, but while there are some questions in which the system offers a clue to the answer as a help to students, there are others that consist of just the questions that the students have to answer.

Based on the source sentence Gabonetan karroza-desfile ikusgarria egiten dute (At Christmas, an amazing float parade is held), ArikIturri is able to generate at least two different short answer questions, as examples II.2.6 and II.2.7 demonstrate.

Example II.2.6 (Short answer example — with clue)

NOIZ *egiten dute karroza-desfile ikusgarria? (Gabon)*
WHEN *is an amazing float parade held? (Christmas)*

Example II.2.7 (Short answer example — without clue)

NOIZ *egiten dute karroza-desfile ikusgarria?*
WHEN *is an amazing float parade held?*

Therefore, in our system, short answer questions can be used to test various knowledge or topics. The main reason for providing a clue (example II.2.6) is that these types of item are focused on grammar. In the given example, students have to understand the question correctly, but more importantly, they have to provide the correct word form of the given lemma (the clue). As grammar is the topic of the question, the item is focused solely on the declension case, meaning that students do not also have to guess the correct lemma of the answer. This way, we can ensure the validity of the item. Although we are aware of the fact that this type of item (the ones that contain a clue) could be seen as word formation items, we classify them as short answer questions.

When the aim of the item is to test a higher-level knowledge, such as reading comprehension items, the generated short answer questions do not offer any clues, showing just the question, as in example II.2.7.

In both cases, the correct answer is stored together with the rest of the information relating to the item as an instance of the question model. However, the correct answer can be seen as one possible answer to the question, as short answer questions could have more than one correct answer. It is up to teachers to decide whether or not to mark the students' responses based only on the source option.

Finally, it is important to mention that the generation of short answer items has not been evaluated as an item generation task but as a task to generate interrogative statements from declarative ones. More specifically, the evaluation focused on how well the system creates the corresponding wh-word[9] based on the morphological information regarding the topic of the items. In addition, the evaluation measured the system's performance in transforming a sentence into its corresponding interrogative form. The experiments which focus on this task are explained in section V.2. In fact, this is a challenging task that has attracted a number of researchers over the last three years. Moreover, in 2008, the first Workshop on the Question Generation Shared Task and Evaluation Challenge (QGSTEC) (Nielsen, 2008) was held and this was the starting point for this ever-increasing community.

II.3 Topic

As we have previously mentioned, for us, the term topic refers to the concept that students have to work with. This term comprises a range of concepts, from the simplest unit of work to the most complex. For instance, the topic of an item could be the conjugation of a concrete verb or reading comprehension. Almost all of the related works are focused on learning the English language. Among others, Chen *et al.* (2006) focus on grammar tests while Liu *et al.* (2005) and Sumita *et al.* (2005) work with vocabulary. In addition, the approach of Hoshino and Nakagawa (2007) generates items designed to deal with more than one topic. Mitkov *et al.* (2009) also produced English items, but this study is focused on a specific domain rather than language learning. Finally, Nikolova (2009) created items from Bulgarian texts; this is one of the works that do not address English items.

Within any learning programme, documents relating to the syllabus and

[9] In this dissertation, we refer to interrogative words as wh-words.

curriculum are used. The syllabus comprises the topics to be covered in a course so that different topics are learnt. In contrast, the term curriculum is used to refer to a wider programme or plan. A curriculum is prescriptive, and is based on a more general syllabus which merely specifies which topics must be understood and to what level in order to achieve a particular grade or standard.

These kinds of programmes are available for the Basque and English languages as well as for different domains. In this dissertation, we focus on three different scenarios: (a) Basque language learning; (b) English language learning; and (c) scientific vocabulary learning. Therefore, the system works with different topics that are part of the syllabus of these three scenarios.

The language learning process in Europe was standardised some years ago and the “Common European Framework of Reference for Languages: Learning, Teaching, Assessment” (CEFR) document was published in 2001. The CEFR provides a basis for the mutual recognition of language qualifications, thus facilitating educational and occupational mobility. In addition, the Council of Europe developed a European Framework with common reference levels of proficiency. There are six levels, which constitute an interpretation of the classic division into basic, intermediate and advanced learning. Basic users are grouped into levels A1 and A2, intermediate users into B1 and B2 and, finally, proficient users into C1 and C2. In our experiments, we focused mainly on the C1 level learners who:

> Can understand a wide range of demanding, longer texts, and recognise implicit meaning. Can express him/herself fluently and spontaneously without much obvious searching for expressions. Can use language flexibly and effectively for social, academic and professional purposes. Can produce clear, well-structured, detailed text on complex subjects, showing controlled use of organisational patterns, connectors and cohesive devices (CEFR, 2001).

In this study, English verb tests (cf., section VI.3) and Basque declension and verb tests (cf., section V.3.1) were generated for C1 level learners. However, the experiment which focused on determiner tests (cf., section V.3.2) was targeted at A1-A2 level learners.

Apart from the CEFR document, there are more specific documents which provide the details for each language that were taken into consideration when designing the experiments.

Regarding Basque language learning, the document "Helduen euskalduntzearen oinarrizko kurrikulua"[10] (HEOK) (HABE, 1999) specifies the Basque language learning process for adults. This document is also placed within a broad plan. Within the document, the process of learning the Basque language is divided into four levels. The first level offers a strong basis to the learners, which is then studied in greater depth in the second level. The third level is similar to the level required to obtain the Euskararen Gaitasun Agiria (EGA): level C1 in the Common European Framework. Finally, the last level, focuses on professional, specialised and scientific Basque.

Each defined level in HEOK has different objectives and content that are established for different skills: reading, writing, listening and speaking. In this work, we have generated items that take into account the morphosyntactic aspects that learners have to acquire during their learning process in order to deal with the aforementioned skills. For this reason, we worked with two different approaches: the generation of items from a Basque language learning corpus (cf., section V.3.1) and the generation of items from a Basque language learner corpus (cf., section V.3.2).

As regards English language learning, the system is also focused on the C1 level and on the Academic Word List (AWL) (Coxhead, 2000). The AWL was made primarily so that it could be used by teachers as part of a programme preparing learners for tertiary level study or used by students working alone to learn the necessary words to study at university. In the case of the English language, the system creates multiple-choice items for learning vocabulary. For this reason, we explored two different corpora: a general corpus and a specific one (cf., section VI.3).

The system not only generates items in a general domain, but also in a specific domain. In order to work with the science domain in the Basque language, we studied the Ostadar[11] project. This project has been designed for the management of the curriculum for the four grades of the obligatory secondary education (OSE) in the Basque Country. The project comprises teaching material from six different subjects: Basque and literature, Spanish and literature, mathematics, social science, natural science and music. These six subjects are taught by means of paper materials, audio-visual media, multimedia and the Internet. In our approach, the system concentrated on the natural science domain due to the possibility of making use of a

[10] The basic curriculum for the process of learning the Basque language for adults.

[11] http://www.ostadar.net/

specialised corpus and real texts in order to generate multiple-choice items. That is to say, the system has been designed to create items in order to test vocabulary from real texts (cf., section VI.4). In comparison to the rest of the generated items, the items relating to the science domain are presented as part of the text as a whole, as this type of task is proposed within the curricular project defined for the natural science domain within Ostadar. In addition, the experiments presented in section VI.4 have been evaluated in a real scenario in which a considerable number of students took part in the evaluation process. Thus, it has been possible to simulate the entire testing process.

In conclusion, depending on the scenario, the input corpus and the defined heuristics, the end-users of the generated tests vary. In fact, we believe that if the necessary corpus and information to define new heuristics are available, new scenarios in which different types of student are the participants could be created with relative ease.

II.4 Architecture

The option of offering a wide variety of scenarios in which different types of item are generated has to be supported by a solid architecture. For this reason, the process of adding a new type of item, feature or heuristic has to be easy to complete. A modular architecture allows this type of easy-to-implement process. In addition, the choice to produce items in different languages has to be foreseen in order to design a system in which this feature is taken into account. Therefore, both multilinguality and modularity are features of ArikIturri.

II.4.1 Multilingualism

Although there are Web-based applications with a multilingual interface as well as the option of completing tests in different languages (Kerejeta *et al.*, 2005), the option of automatically generating items in different languages is not usually offered. In contrast, as we will show in chapters V and VI, ArikIturri is multilingual, that is to say, the system is able to generate items in different languages.

As the architecture of the system is modular (cf., section II.4.2), the system is well-suited to the addition of a new language. In this dissertation, we demonstrate the viability of the system to generate items in two languages: Basque and English. Thus, the experiments herein will prove the ability of the system to generate items in both languages, as well as the ease of doing so. Broadly speaking, the only requirement for adding a new language in ArikIturri is the availability of the required NLP tools and the input corpora (Aldabe *et al.*, 2009).

II.4.2 Modularity

One of the paradigms that allows the implementation of a system which can adapt easily to new characteristics is the object-oriented programming (OOP) paradigm. Objects are the basic entities in OOP, and they are composed of data and procedural instructions. A class is a collection of objects and is used to make instances of objects. In general, an object model is based on **abstraction**, **encapsulation** and **inheritance**.

Abstraction refers to the act of describing the essential properties of an object without including details, in other words, describing the conceptual generalisation of the attributes and properties of a set of objects. For instance, the *Criterion* class of the class diagram presented in Figure II.3 can be seen as an abstraction of the commonality between all of the different types of criteria (*PICriterion*, *CSCriterion*, *AFICriterion*, *DGCriterion*, *IGCriterion* and *QRCriterion*).

Encapsulation concerns two ideas: data protection and the separation of external functionality from internal implementation. Thus, from the user's point of view, an object can be seen as a black box. For example, and based on the same *Criterion* class, the class is responsible for establishing a criterion based on certain features. In order to do so, the essential attributes and methods must be publicly available. Thus, different criteria are applied based on the language, question type, level, topic and heuristic so that the values of those attributes are established, and by means of the public function *getCriteria*, the process of obtaining the criterion is carried out. The rest are internal or private attributes defined according to the correct running of the class.

Inheritance or hierarchy is the feature by which objects can acquire the properties of objects from another class, in the way that children can inherit characteristics from their parents. In the case of the *Criterion* and *PICriterion* classes, for instance, all of the methods and attributes of *Criterion* are acquired by the child class *PICriterion*, after which new methods can be added or the inherited ones redefined.

The idea of encapsulation provides **modularity** to the implementation. Thus, modularity refers to the decomposition of a system into separate modules. ArikIturri is defined in a modular way, thereby allowing us to implement and update each module in an independent way.

In addition, two major advantages of OOP are reusability and portability, and these are, in fact, two of the aims of our system. Reusability refers to the option of reusing the program code by changing a few parameters. Offering ArikIturri under a GPL license enables the free distribution of the system and allows the scientific community to reuse the code. Portability relies on the idea that by changing a few characteristics, the system could be used on different platforms. Although the system was tested in a Sun server under Solaris, all of the configuration parameters are offered separately so that they can be easily adapted to other platforms.

Figure II.2 shows the main modules of the architecture. In our approach, the generator uses as the main input parameter a set of morphologically and syntactically analysed sentences (the tagged corpus), represented in XML. Based on the rest of the parameter specifications, the system transforms the input corpus into items, which are also represented in XML. As this is a modular process, the system obtains results in the intermediate steps which are also represented in XML.

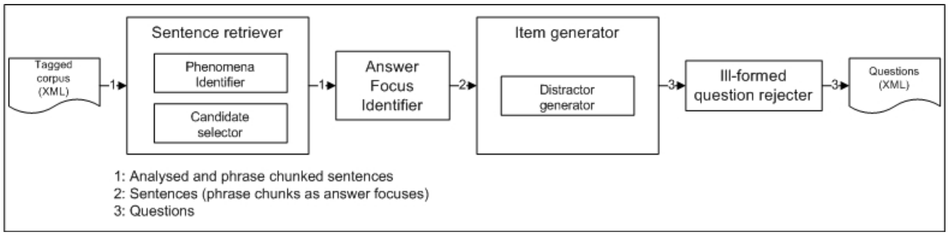


Figure II.2: The architecture of ArikIturri

Therefore, the process of generating test items can be summarised as follows (Aldabe *et al.*, 2006): based on the parameters' specifications, the *sentence retriever* module selects candidate sentences from the source corpus which has been designated as the source. In the first step, it selects the sentences where the specified topic appears. Then, the *candidate selector* module, based on the defined criterion, selects the candidate sentences.

Once the sentences are selected, the *answer focus identifier* marks out some of the phrases as focal points for the answers depending on the information contained within them. Then, the *item generator* creates the questions in accordance with the specified exercise type. This is why, this module contains the *distractor generator* sub-module.

As the entire process is automatic, it is probable that some of the questions will be ill-formed. For instance, in the Basque language, there are some words which have the same surface form despite the fact that the deep morphological form is different. For this reason, we included the *ill-formed question rejecter* module in the architecture. For instance, this can occur if there are an indefinite or plural number of nouns. Let us imagine that there is a heuristic which generates distractors which change the number of the key, and so as the number of the key is indefinite (egunetan (day)), it would generate as distractors the singular and plural forms of the key as example II.4.1 presents.

Example II.4.1 (Ill-formed example)

Edozein .... bueltatuko dira etxera
(They will return home any ....)
a) egunean (distractor) (day)
b) egunetan (key) (day)
c) egunetan (distractor) (days)

In the given example, the key and one of the generated distractors, the plural form of the key, refer to different numbers but the word form is the same. Among other verifications, this type of repetition is checked by the *ill-formed question rejecter* module.

We have already mentioned that the system obtains not only complete items but intermediate results. Obtaining in-between results allows the system to avoid the repetition of some processes. Let us imagine a multiple-choice test that has been generated in order to deal with verbs in the Basque language. After this generation, a teacher wants to test the students again using the same topic and the same sentences, but this time using FBQs. The system would start from the item generator module instead of the sentence retriever module (see Figure II.2), due to these intermediate results.

Most of the works which focus on the automatic generation of items do not present an explicit architecture, but instead present steps for carrying out the process. Nevertheless, there are some differences and commonalities between these steps and our system. For example, to our knowledge, our system is the first to integrate an answer focus identifier module.

Smith *et al.* (2009) present a complete architecture of the TEDDCLOG system, but there is a difference between their generation process and ours. They generate the distractors first and then select the sentences. In ArikIturri, the sentence-selection task precedes the distractor-generation task, because depending on the topic, the information in the sentence can be used to produce the distractors. Both proposals are different ways of contextualising distractors.

Equally, Liu *et al.* (2005) and Sumita *et al.* (2005) follow the same procedure as ours in order to generate the questions. While Liu *et al.* (2005) do not incorporate any processes intended to discard ill-defined items, Sumita *et al.* (2005) also include a module designed to reject questions based on the Web. Therefore, the process presented by Sumita *et al.* (2005) can be seen as being the most similar to ArikIturri.

Apart from the modular representation of the system presented in Figure II.2, as ArikIturri has been implemented by means of OOP, it is also possible to present the functioning of the system by means of different classes. These classes are shown in Figure II.3.

The main class is called *QuestionGenerator*, and it is responsible for generating the test items. For this purpose, it makes use of the following classes: *SentenceRetriever*, *AnswerFocusIdentifier*, *ItemGenerator* and *QuestionRejecter*, which can be seen as the main classes of each module.

We have already mentioned that the generator uses as input a set of morphologically and syntactically analysed sentences (the tagged corpus), represented in XML, in which chunks are marked. However, it is not compulsory to provide the system with a previously analysed corpus. Before starting the generation process, ArikIturri checks the state of the source corpus. Thus, in cases in which the corpus is composed of raw texts, ArikIturri makes use of the *Preprocess* class to obtain the analysed corpus.

In the case of Basque language, the *Preprocess* class uses *Ixati* (Aduriz *et al.*, 2004) to analyse the texts (cf., section IV.1.1.1). The output of this analysis is an XML file in which chunks of the sentences are marked together with their morphological and syntactic information.

For English texts, Connexor Machinese Syntax (Tapanainen and Jarvinen, 1997) is used (cf., section IV.1.1.3). The output of this analysis is a syntax tree in which chunks are not represented explicitly. To this end, we have implemented an extra module for the English version which takes the

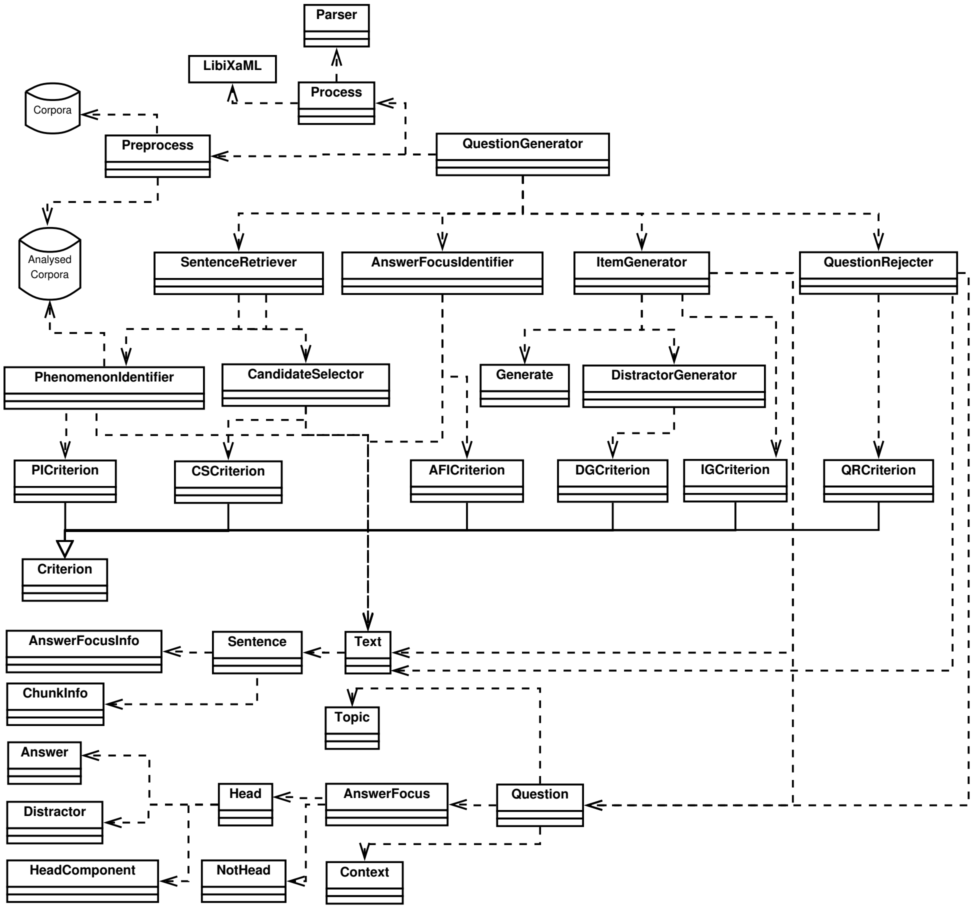


Figure II.3: The classes of ArikIturri

output of the parser and produces the chunks needed for the generation of test items (Aldabe *et al.*, 2009).

In the following sections, the main modules are explained more extensively. For each module, we present its main purpose, the main methods used to carry out the task, the NLP tools used, the necessary linguistic information and the data.

II.4.3 Sentence retriever

The main purpose of the *sentence retriever* module is to select candidate sentences based on the parameters' specifications. For this reason, the main class *SentenceRetriever* employs the *PhenomenonIdentifier* and *CandidateSelector* classes. The former identifies the sentences in which the relevant topic appears. In addition, it stores the obtained results in an XML file, as well as in a new object called *Text*. The latter, based on previously established criteria, selects the candidate sentences to be part of the items. Therefore, after the phenomenon identifier is applied, all of the sentences which refer to the relevant topic are offered, and then, after the candidate selection task, only the required number of items are proposed.

First of all, the *PhenomenonIdentifier* class establishes the candidate selection process. Depending on the topic that the experts want to work with, the process varies. Thus, the system consults the criteria established in the *PICriterion* class, based on which the identification process is carried out.

For the time being, and based on the defined experiments, we have established two main groups of criteria. On the one hand, candidate sentences must contain as part of their phrases the morphological linguistic phenomenon that the topic refers to. That is to say, the selection is based on the morphological information contained within the sentences. On the other hand, candidates must be selected based on the semantic information contained within the words.

When the morphological information is the established criterion, the process of detecting the corresponding phrase is fairly simple, as the process comprises only the detection of adequate information. Therefore, in this case, the adequacy of the sentence retriever depends on the accuracy of the morphological disambiguation. In contrast, when dealing with semantic features, the process of detecting the corresponding sentences can be more complex. Thus far, we have studied two approaches: making use of a list of meaningful terms and detecting automatically the meaningful terms in sentences.[12] In the case of a list of terms, the system has to detect the terms in the corpora being analysed. When the system has to extract the meaningful terms from the sentences by itself, it can do so at the sentence level, text level or corpus level. In this approach, we are interested in working with the terms that are important at the text level, as we aim to create tests which comprise an entire text with their items. In order to detect such terms in an automatic way, the system needs to incorporate a term extractor. A particular instance of this task is shown in section V.2 in which the meaningful terms are numerical

[12] Another approach relating to the use of semantic information involves basing the selection on semantic role labelling information, for instance.

entities.

The functions defined in order to detect candidates are language-dependent, because the analysers used in our approach for Basque and English are different. In fact, nowadays, it is not possible to analyse both languages with the same analyser. Thus, the system makes use of two different libraries throughout the entire process in order to deal with language-dependent duties. These two libraries have been grouped into the *Process* class so that all of the libraries of this type may be together as a helping class.

In the case of the Basque language and its linguistic information, the system uses LibiXaML (Artola *et al.*, 2009) to work with the analysed corpora and to detect the necessary phrases. LibiXaML (cf., section IV.1.1.1) offers the necessary types and operations to manipulate the linguistic information in accordance with the model. This class library has been implemented in C++ and the annotation model relies on XML technologies for data representation, storage and retrieval. For the implementation of the different classes and methods, it uses the Libxml21 library.

In the case of the English language, we implemented a SAX parser to identify the terms to address, and to work with the particularities of this language in general. For this purpose, an object called a *Parser*, which is responsible for parsing the input corpora, was created. This object can be seen as a library of the system.

While the candidate selection process is carried out, the information obtained is collected in two different components. On the one hand, it is stored in an XML as the intermediate result. On the other hand, it is stored in the *Text* class (see Figure II.3). The *Text* object and its components are used throughout the entire generation process to store the information obtained in each step. Conceptually, the *Text* represents one text of the source corpus that is the source of at least one item. Each sentence of the text that is used as part of an item is represented by the *Sentence* class which represents the sentence and distinguishes the information relating to the answer focus from information relating to chunks. For this purpose, it makes use of the *AnswerFocusInfo* and *ChunkInfo* classes. All of these classes are helping classes, designed to store the source information during the automatic process in order to ensure easy access.

Once all of the candidate sentences are detected, the *CandidateSelector* class is responsible for selecting a specific number of sentences based on the criteria established in the *CSCriterion* class. We have not studied all of the possible criteria in depth. Therefore, ArikIturri, at present, selects candidates at random. When a more detailed study is conducted, the resulting criteria could be easily added to the *CSCriterion* class. In fact, the addition of a new criterion to the system only requires the corresponding criterion file to be added and the corresponding function to be updated.

All of the main classes of the architecture are based on a particular "criterion" object (*PICriterion*, *CSCriterion*, *AFICriterion*, *DGCriterion*, *IGCriterion* and *QRCriterion*) in order to allow easy adaptation to new features. Although each criterion has its own particularities, it is clear that some general features can be established. That is why the *Criterion* class is the parent class of all of the classes. Children classes inherit all of the methods and data of the *Criterion* class. With regards to the *PICriterion* and *CSCriterion* classes, these methods form the basis for establishing the way to select sentences in this particular module. In the following subsections, the rest of the children of the *Criterion* class are described and explained.

II.4.4 Answer focus identifier

Conceptually, once the sentences in which the phenomenon appears are detected, the system has to identify which components of the sentence are the ones which are required for working with the topic. With this purpose, the sentences will be marked not only with linguistic information but also with information relating to the topic. The *answer focus identifier* module, as its name expresses, is responsible for identifying the *answer focus*. For us, the answer focus needs to be understood as the minimum amount of information required in order to deal with the topic. The *AnswerFocusIdentifier* class is responsible for this process of identification.

Thus far, we have distinguished two types of marking: word-level marking and phrase-level marking. This distinction is necessary because the information required to specify each topic varies.

Based on our experiments, the declension cases, verb forms (cf., section V.3.1) and vocabulary (cf., sections VI.3 and VI.4) topics in question were marked at the word level. For instance, if the topic is related to the correct use of the sociative case, the minimum amount of information required in order to deal with it and consequently to generate the corresponding item is the word containing the corresponding suffix to mark the noun as sociative. Similarly, if the topic is related to vocabulary and the term is composed of one word, the information required is the word itself, independent of the process used to detect it.

In contrast, the items for topics which are related to the use of determiners (cf., section V.3.2) contain answer focuses with more than one word, that is, the marking is carried out at the phrase level. In Basque, the order of the elements that constitute the noun phrase (NP) is fixed. In addition, the determiner, in general, appears at the end of the NP, in some cases agglutinated to a word, and it takes the entire NP as its complement, constituting the determiner phase (Laka, 1996). Thus, there are cases in which the linguistic features of the other components of the phrase are necessary in order to deal correctly with the determiner topic. For instance,[13] based on the source sentence Lagun asko joan ziren (Many friends went) it is possible to generate error correction example II.4.2.

Example II.4.2 (Error correction example — Answer focus)
Asko lagun joan ziren
Friends many went

Example II.4.3 (Error correction example — Answer focus)
Euskal Herria nazioa bat da
The Basque Country is a one nation

Example II.4.3 also shows an error correction item which deals with the correct use of the determiner. In this case, the information contained in the phrase is needed because a NP cannot take either two determiners or a determiner and a quantifier (Uria, 2009), and this information is given at the phrase level when detecting the determining phrase nazio bat (a nation) in the source sentence Euskal Herria nazio bat da (The Basque Country is a nation).

Another particular case of this phrase marking is studied in section V.2. This experiment focuses on the transformation of declarative statements into interrogative ones, during which meaningful terms are automatically detected and classified. These meaningful terms are numerical entities that contain between 1 and n components. In cases in which these entities are used as the topic of the items, the answer focus would consist of more than one word.

Although we have not addressed them, other examples of phrase marking are topics which deal with the agreement in phrases or even topics relating to the order of the words. This last topic could be seen as a form of phrase marking that can cope with more than one phrase. Although this type of

[13] Examples extracted from (Uria, 2009).

marking could be considered as sentence-level marking meaning that the answer focus should contain more than one phrase or chunk, in such cases, the system would distinguish two different answer focuses that would be related to one another (cf., section III.3.1.2).

All of the information obtained in this step of the generation process is collected into the *AnswerFocusInfo* class to be used when generating the questions. However, it is necessary to clarify that as the analysis of the input corpus is carried out at chunk level, the corresponding component of the answer focus in the question model (cf., section III.3.1) contains two elements: one to represent the information required in order to deal with the topic (the *Head* element) that depends on whether the marking comprises one or more words, and the other one, the *notHead* element, that represents the rest of the chunk (in cases in which there are more elements).

II.4.5 Item generator

Once the sentences, chunks and answer focuses are identified, the *Item generator* module is responsible for creating the questions. For each item, the module creates a question instance. Although the question model is explained in depth in chapter III, in the following, we provide a brief idea of what a *Question* is in order to enlighten the reader.

Conceptually, a question contains all of the information relating to a particular item. Thus, apart from components such as the stem, the key and the distractors, a question provides information relating to the generation process. Therefore, a *Question* (cf., section III.3.1.1) is composed of three main components: the *Topic*, the *AnswerFocus* and the *Context*. As previously mentioned, the answer focuses are the chunks[14] of the sentence in which the relevant topic appears. The rest of the chunks of the sentence are collected in the context.

We have also already explained that the list of words[15] in which the topic in question appears is represented by means of the *Head* element. More specifically, the *Head* component is composed of three elements: the *Answer*, a list of *Distractor* elements and a list of *HeadComponent* components. The latter two elements are part of the *Head* and depend on the topic and question type, while the *Item generator* module is responsible for identifying the need

[14] The minimum information required in order to deal with the topic.

[15] From *one* to *n* words.

for these components as well as for generating them.

Therefore, depending on the question type, before generating a question, the module has to create additional information in order to complete the items. These new components are not part of the source sentence, so the *ItemGenerator* generates them based on the previously defined criteria. Thus, like the preceding modules, the *ItemGenerator* is also based on the identification of the components of the criteria defined in the *IGCriterion*.

Depending on the type of question, the established criteria specify that a process to generate at least one distractor is required. As a distractor is defined as the list of words which are incorrect in the given context, the *Distractor* element is always linked to the *AnswerFocus*. Apart from the list of words, the *Distractor* class also contains the corresponding linguistic analysis as well as the heuristics used for creating each distractor. We consider that, conceptually, the process of generating the distractors is significant enough to be considered as a sub-module of the system. This sub-module is called the *distractor generator* and is explained in more detail in section II.4.5.1.

In addition to specifying the criteria to be followed in the generation of the distractors, the *IGCriterion* class also contains the criteria to be used in the generation of the *HeadComponent* elements. Conceptually, the *HeadComponent* collects the specific information relating to the question type that is not part of the source sentence and that is not a distractor. We have worked with one element that is not part of the source sentence but that is directly related to the topic or question type: wh-words.

For instance, when the topic of an item aims to deal with the vocabulary of a particular subject, one way of testing such knowledge is by asking about certain concepts after a passage has been read. Thus, the stem of this type of item can comprise an interrogative statement, meaning that the corresponding wh-word has to be generated based on the term to be tested: the answer focus. If the *IGCriterion* establishes such a criterion, the *Item generator* module is responsible for automatically generating the wh-word and creating a new *HeadComponent* element. The experiment presented in section V.2 explains how this process is carried out.

II.4.5.1 Distractor generator

As previously mentioned, in cases in which an error correction question or an MCQ is generated, the system has to create distractors. The module responsible for this generation is the *distractor generator* and it is implemented by means of the *DistractorGenerator* class.

The *DistractorGenerator* class is based on the criteria stored in the *DGCriterion* class in order to generate distractors. That is to say, the previously established criteria are the starting point for generating the distractors and applying the desired heuristics. These heuristics are based on knowledge acquired manually or automatically and can represent anything from the simplest to the most complex generation process.

Based on the criteria defined in the *DGCriterion* class, the *distractor generator* module has to first establish the heuristic and then generate a new word. In order to carry out the aforementioned generation process, the system makes use of the *Generate* class. This class can be seen as a class library in which all of the different generation processes are grouped.

Once all of the components of the item are generated, the *ItemGenerator* has to create a question to represent all of the stored information. For this purpose, it takes into account all of the necessary information that contains the *Text* class, as well as the components generated by this module.

II.4.6 Ill-formed question rejecter

We have already mentioned that the process of generating the question is completely automatic. Due to various factors, it is possible for incomplete or ill-formed questions to be generated. That is why we have included the *ill-formed question rejecter* module into the architecture, to detect and reject this type of question. In order to do so, the system contains the *QuestionRejecter* class.

There are various reasons for discarding a question, and these reasons can be categorised into two main groups: reasons for discarding an item and reasons for discarding a distractor. In addition, depending on the defined criteria, discarding a distractor does not necessarily entail the rejection of the item as a whole. Thus, the *QRCriterion* is responsible for establishing these criteria.

As regards the reasons for discarding an item, based on our experiments, two different reasons have been identified: one the one hand, to prevent the repetition of question which already exist, and on the other hand, to reject items that start with a blank. The latter reason was identified after an experiment was conducted with editors (cf., V.3.1.2).

Regarding the reasons relating to the generated distractors, we have encountered different scenarios. First of all, there are cases in which it is not possible to generate as many distractors as required. We have detected that in some cases, the system is unable to produce the expected number of distractors due to linguistic characteristics. For instance, there is no plural for proper nouns, and so it is not possible to generate automatic distractors to work with in such cases. Let us imagine that the system is dealing with the sociative case (the topic), and one of the defined heuristics changes the number of the head of the answer focus. If the system tries to apply this heuristic in the source sentence Kubarekin zerikusia duen guztia atsegin dut (I like everything which is related to Cuba), the system would not be able to offer any candidate distractor for the head Kubarekin (to Cuba), because it is a proper noun. This type of reason would lead to the rejection of the entire item.

Example II.4.4 (Problematic item)
.... joan nintzen etxera. - I went home ...
a) Anaiarekin - correct answer - sociative singular - with my brother
b) Anaiekin - correct answer - sociative plural - with my brothers

Second, there are cases in which the generation of a candidate distractor results in the generation of an additional correct answer. Example II.4.4 shows an item in which the correct use of the number is tested with the source sentence Anaiarekin joan nintzen etxera (I went home with my brother). If the answer focus is Anaiarekin (with my brother) and the heuristic used is to change the number of the correct answer, then one of the candidate distractors may be Anaiekin (with my brothers). The problem arises from the source sentence, because the lack of context makes the candidate distractor a possible correct answer.

Although we have detected such problematic cases, it is not possible to identify such behaviour by means of the NLP tools available. However, we do believe that a criterion based on the role information of verbs could help in such scenarios. In any case, this reason does not necessarily result in the rejection of the entire item, but merely requires the item to be updated when it is given to experts.

Finally, there are cases in which the distractor generation task produces the same surface word form distractors, even if these distractors have different deep forms. Example II.4.5 shows an MCQ type which was rejected because of the fact that there are two identical distractors, i.e., (b) and (c), for different inflected forms. The choices are answer focuses in which the head of the answer is in bold.

Example II.4.5 (Automatically rejected item)
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 names of the political party*
*c) alderdiaren **izenetan** - innesive indefinite - in the name of certain political parties*
*d) alderdiaren **izen** - lemma - name of the political party*

There are some cases in which it is possible to reject just some components of the item and to replace them with new ones. For instance, when working with semantics, if ArikIturri is able to detect that a candidate distractor is a correct answer and consequently to generate a new valid one, there is no need to discard the item.

In conclusion

In this chapter, we have presented the main features of ArikIturri, a system which is modular, multilingual and independent of any application that uses its items. The system is able to produce FBQ, word formation, MCQ, error correction and short answer questions. In addition, ArikIturri deals with several topics. In a Basque language learning scenario, the system generates items that take into account the morphosyntactic aspects that learners have to acquire during their learning process. In the case of English language, ArikIturri creates MCQs for learning vocabulary. Finally, the system also explores the generation of items in a specific domain. For that, ArikIturri focuses on the natural science domain to test Basque science vocabulary.

CHAPTER III

Question model

In this chapter, we present the question model which has been designed in order to describe the items generated by ArikIturri. We describe the model underlying the system which is flexible and general. In addition, this model contains the information relating to the generation process of the items.

III.1 Motivation

One of the main motivations to define standards in the e-learning domain is to make content sharable, reusable and accessible across different LMSs. This way, their independence from any specific LMS is ensured. In addition, applications that are developed within the e-learning domain should guarantee interoperability at different levels and implementation according to standards such as Shareable Content Object Reference Model (SCORM) (ADL, 2009), IEEE Learning Object Metadata (LOM) (IEEE, 2002) and IMS.[1] Thus, the test items generated by ArikIturri should be represented in a standard way in order to ensure their durability, portability and reusability.

The IMS QTI (IMS Global Learning Consortium, accessed 2010) is a standard for representing questions and test data and their corresponding results, enabling the exchange of data across different LMSs.

[1] http://www.imsglobal.org/

> The IMS QTI specification has been designed to support both interoperability and innovation through the provision of well-defined extension points. These extension points can be used to wrap specialized [sic] or proprietary data in way that allows it to be used alongside items that can be represented directly (IMS Global Learning Consortium, accessed 2010).

Therefore, it is possible to extend QTI with other XML namespaces like MathML[2] for formulae or to add namespaces of one's own design. As the portability of the generated items is one of our goals, it makes sense to offer the questions represented by the QTI standard. However, the main objective of this dissertation is to represent not only the items but also their corresponding topical information, as well as the heuristics used for their generation. As we will explain later, this additional information should be represented as an extension point. Therefore, we first defined our own model and then developed an adaptation of the model to the QTI standard. A two-step process was followed because the main objective was the representation of the information concerning the generation process and not the assessment process of the items. Thus, we first established the elements concerning the items to be represented. Second, as other authors have done before us (Guzmán *et al.*, 2005), we provided our own specification. Finally, after analysing the elements of the question model, we supplied a QTI extension to offer to a wider community, as others have done before (Mavrikis, 2005).

As the purpose is to define a general model, the concepts concerning the representation of a question are not related to a specific question type. The concepts have been defined independently from the question type. In addition, the characteristics of different question types have been taken into account in order to improve the generalisability of the model and its components. In our model, a question is a sentence or clause in which the topic which the student has to focus on appears and is represented as part of the text as a whole.[3] Therefore, based on the defined concepts, the question model comprises all of the components of the items and, depending on the question type, the question will have certain specific components.

Regarding the heuristic information, there are different reasons for supplying the items with this information. By means of this additional infor-

[2] http://www.w3.org/Math/

[3] Although our items are generated from texts, there is also the option of representing isolated items, as we will see later on.

mation, experts can consult the heuristics in order to better understand the generation process. Once they interpret these heuristics, they can improve the system by providing it with feedback. For this reason, we consider it to be indispensable to implement an application with a user-friendly interface in which the information regarding the generation process presented in the question is clearly represented. The application should be capable of providing the information in a guided way in order to facilitate users' understanding. Moreover, the option of providing feedback should be parameterised in order to facilitate the experts' work.

From a student-oriented perspective, information about the generation process provides hints to guide students in a technology-enhanced learning system. For instance, a student-oriented application which contains a learner model can consult the mistakes previously made by a student and propose some items based on this information. Let us imagine that the topic of an MCQ is the correct use of the preposition of within an English grammar test. In order to deal with the topic, ArikIturri comprises, among others, a heuristic which generates a distractor which changes the preposition of (the key) into the preposition from. Some students may have problems distinguishing between the two (of and from) prepositions. Thus, these previously detected students could complete a test focused on grammar in which some of the distractors would be generated based on these two prepositions. The heuristics could also be used for the diagnosis itself. For example, if a student tends to choose the from preposition instead of of, the system could detect this repetitive behaviour thanks to the heuristic information. This way, the system could perform two actions. On the one hand, it could store this information for future sessions. On the other hand, the system could alert experts to this behaviour.

In general, the items generated by ArikIturri are offered to experts and students (after teachers have overseen them) by means of different applications. For this purpose, the system provides an export process which is an easy task thanks to the fact that the questions are represented in XML in a structured way. In the following sections, the main characteristics and the structure of the model are broadly explained.

III.2 General and flexible

When describing the model, we did not focus on just one type of question, but on several types. More specifically, various selected-response and constructed-response items were analysed when defining the model. The model describes general concepts which are common to all question types.

One of the aims of the model is to allow the representation of the information relating to the generation process. That is to say, the model provides a structured and general way of representing the linguistic information relating to each item. For that reason, the model permits the representation of the information relating to any target language. In addition, it is a general model because of its independence from the language and the NLP tools used in the generation process. In fact, multilinguality is also one of the characteristics of the model.

In addition to being a general model, the model has also been designed to be flexible. For this reason, various types of questions can be specified in the same test, which is something that happens in real examinations. The flexibility of the model allows the representation of types of question such as word order and transformation items.[4] However, it was established that all items have at least one correct answer,[5] as our items are produced from texts from which the answers are also derived.

As regards the components of the model, there are different features that make the model flexible. First of all, the model offers the option of having more than one answer focus in the same question. As a consequence, different answer focuses can be related to one another. For instance, the system can generate an error correction item which has two errors focused on the declension of nouns. It is also possible to vary the complexity of the questions by increasing or decreasing the number of answer focuses per item. For example, an FBQ can have as many blanks as the source sentence has chunks. Thus, the higher the number of blanks, the higher the degree of difficulty. Of course, there comes a point at which the legibility of the items is jeopardised due to the number of blanks. It is the system's responsibility to define the maximum number of blanks in each case.

In addition, the order of the chunks in the sentence is interchangeable

[4] The latter items have not been implemented in this dissertation.

[5] Although we have not worked with multiple-answer MCQs, the system can represent this type of item.

in two ways: from the source sentence to the stem and from the proposed stem to the final item. First of all, depending on the question type, there is a need to change the order of the chunks from the source sentence to the stem. For instance, in the Basque language, when writing a wh-question, the wh-word must precede the verb phrase. Thus, it may be necessary to change the order of the verb phrase. As will be explained later, this type of action is represented in the model by means of the *posS* and *posQ* attributes. In addition, there are some languages, such as Basque, in which the word order is partially free, and that is why the model offers the option of changing the order of the chunks of the generated stems by means of the *change* attribute.

Finally, we decided to allow the modification of the components of the stem. For instance, let us imagine a stem composed of some common words. The replacement of the most common words by their corresponding “rare” synonyms would increase the difficulty of the stem, creating a better item to use in advanced levels or grades. There is also the chance of changing the key, the context and the *HeadComponent* elements (cf., section III.3.1). In section VI.4.3.4, the influence of the occurrences of the key in the text will be presented by replacing the key with a synonym. When generating interrogative statements from declarative ones (section V.2), some verb forms need to be transformed in order to create a coherent question.[6]

The model presented in section III.3.1 has been developed using XML. The use of XML makes the import and export processes easy. As one of the aims is to offer the items to the scientific community and as the model itself is not part of a standard, the system provides an export library in order to facilitate the import and export tasks. In addition, the items can also be represented as an extension of QTI (see section III.3.3). The aim of this extension based on QTI is to offer the items to a wider community by means of LMSs.

III.3 Structure

XML is a standard which was first designed to be used over the Internet in order to structure and exchange data, but it is also used in the exchange of a variety of data between applications. In addition, although XML is focused on documents, it is also used to represent data structures. The manner in

[6] This transformation is developed automatically.

which an XML document is formally defined is by defining an XML schema, which specifies the elements and attributes necessary for an XML document to be considered valid. Among the various available schema languages, the main three are Document Type Definition (DTD), W3C XML Schema (XSD) and RELAX NG. Although DTD is probably the easiest option to understand and define, the content models for DTDs are very simple and the data types are limited. Compared to DTD, XSD and RELAX NG are more powerful. As the QTI specification is described in XSD, the model we propose is also described in XSD.

In general, an XML document is well-formed if it has just one root. As our XML schema was designed to represent questions, the root of it is the <questions> tag. By default, attributes are optional, but almost all of the defined attributes in our schema are compulsory. In brief, the head components of the schema are questions which have three main components: the topic, the answer focus and the context. The answer focus is the chunk of the source sentence in which the topic appears and the rest of the chunks are collected into the context. These main components, as well as the rest of the components, are explained in detail in section III.3.1.

III.3.1 Components

The XML schema we propose is focused on items and the information obtained in the process of their generation. As there is no explicit way to represent all of this information by means of QTI, we first defined our own model and then proposed it as an extension of QTI (cf., section III.3.3). To our knowledge, this is the first model that comprises information relating to the process of the generation of the items. As we have previously mentioned, this is very interesting as regards the provision of feedback to the system as well as to the students in the case of LMSs.

When we designed the model, we did not restrict its elements to one specific question type. We studied different question types (FBQ, error correction, MCQ, word formation and short answer) with the aim of providing a general model. As the item types have different characteristics, not all of the elements of the question model are mandatory. In the following, the various elements of the model are explained as well as a complete example of each type of the implemented questions. The XML schema itself is shown in Appendix B.

Figure III.1 shows the question model represented in Unified Modeling

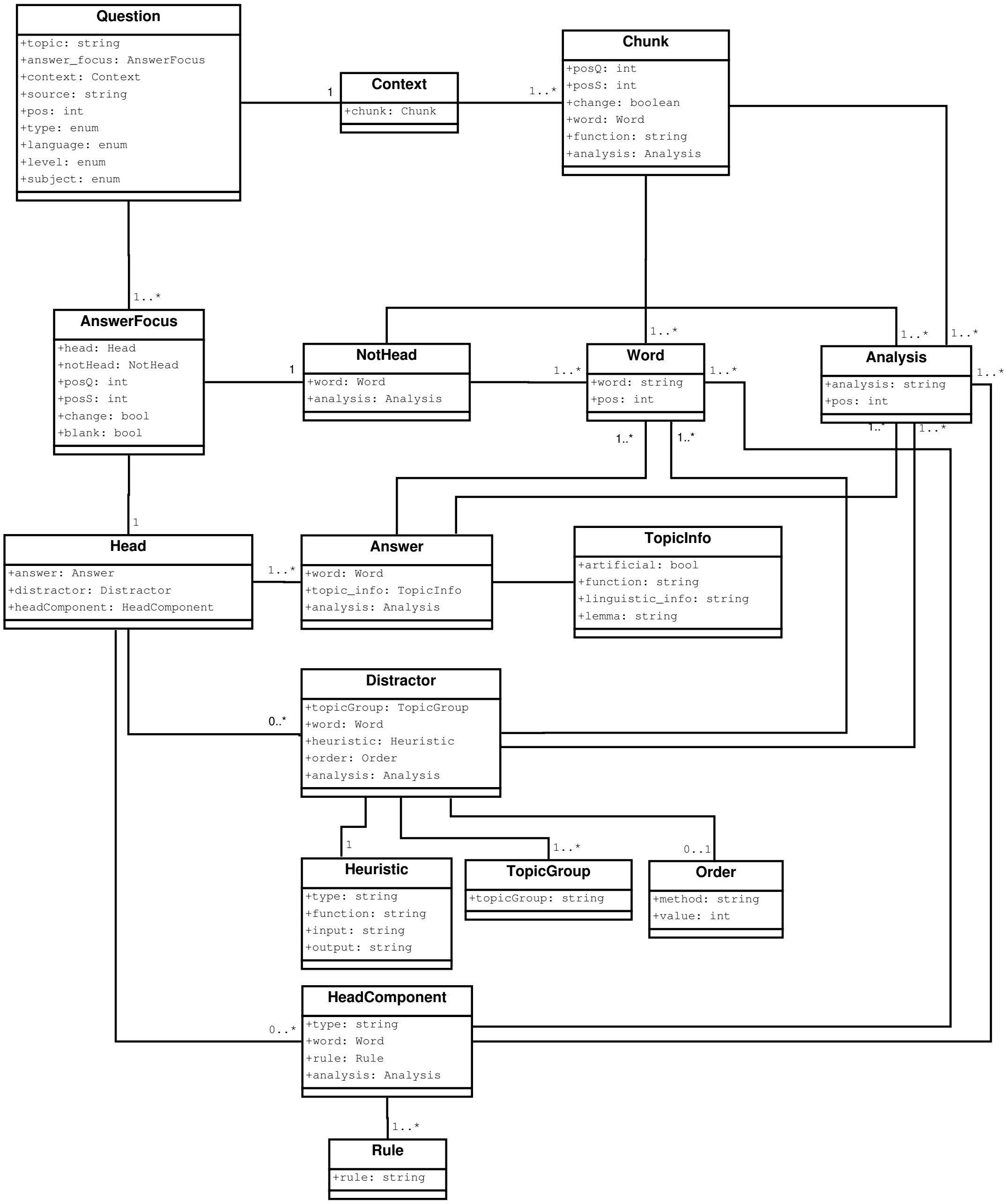


Figure III.1: The question model

Language (UML). In the following sections, starting with the main element of the model —the question— the elements of the model are presented. The concepts relating to each element are explained in a general way and the particularities of each question type are specified when the examples are presented.

III.3.1.1 Question

We have already mentioned that a test comprises a set of questions. A **question**,[7] conceptually, has three main components: the *topic*, the **answer_focus** and the **context** all of which are compulsory elements or attributes. Figure III.2 shows the attributes and elements of a question.[8]

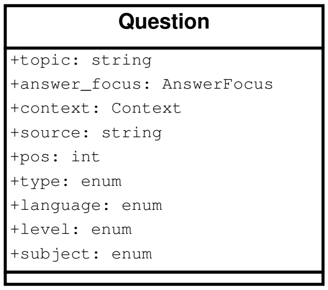


Figure III.2: The question element

As we have explained in chapter II, a corpus is used as input for the system for generating the items. A corpus is a collection of texts stored in a structured way, and the stems of the generated items are part of a particular text. Thus, each item corresponds to a particular source sentence. As regards the sentence, we define the **answer_focus** as the chunk of the sentence in which the topic appears. The rest of the chunks of the sentence are collected into the **context** element.[9] Thus, we propose a general model in which the sentences are represented at the chunk level. Nowadays, some analysers mark the chunks explicitly or implicitly for a wide variety of languages, including Basque (Aduriz *et al.*, 2004) and English (Tapanainen and Jarvinen, 1997).

With regard to the information source, the *source* attribute of the **question** element specifies the source file of the sentence and the *pos* attribute

[7] From now on, the **elements** of the model will be represented in **bold** and the *attributes* in *italics*.

[8] We have not shown the information relating to the sub-elements because it will be shown later.

[9] Both sub-elements are explained in later sections.

refers to the position of the sentence within the source text. In this way, it is possible to present an item within a text as a whole when building a test. However, it is also possible to offer the question as an isolated concept by ignoring the value of this attribute. Although items are thought to be based on a source sentence from a text, it may be possible to generate an item from scratch or from a sentence. If so, it would also be possible to represent the item using our model by establishing the *pos* value at -1. Thus, this value represents the concept of items for which the source sentence does not belong to a specific text and the *source* value is established as being NULL.

The reason for assigning the *topic* attribute to the question element and not to the exercise is based on the fact that different questions designed to address different topics can be part of the same test. In fact, in real examinations such as CAE or EGA different topics are part of the same test. For instance, the Use of English paper in the CAE exam is focused on grammar and vocabulary. Both topics can be represented by the *topic* attribute, as it has been defined in order to represent any topic, from the most general to the most specific. The *topic* attribute allows any value, and so it is the system's responsibility to enunciate the topics in a legible way.

Topic	Experiments
Declension	section V.3.1
Verb	section V.3.1
Determiner	section V.3.2
Vocabulary	sections VI.3 and VI.4

Table III.1: The topic addressed

Table III.1 shows the different values that the system has already established for the *topic* attribute. These are the values with which we worked during the experiments, and so all of them have been implemented and represented in the model. Section V.3.1 deals with Basque grammar tests. More specifically, the items are focused on the correct use of some declension cases and verb forms. Therefore, the *topic* attribute has two possible values in this scenario: "declension" and "verb". In the case of the items presented in section V.3.2, they are also focused on Basque grammar, but on the correct use of determiners. Thus, the corresponding *topic* value for these items is "determiner". Finally, the experiments presented in sections VI.3 and VI.4 are related to vocabulary tests, and so for these items, the set value is “vocabulary”. All of the defined topic values are general enough to be understood at a glance and encompass more fine-grained topics. For instance, the declension topic comprises five declension cases (sociative, inessive, dative, ergative and absolutive).

Depending on the *topic* of the item, the linguistic information required in order to deal with it varies. For topics such as declension and verbs, the linguistic information required is at the word level as the information given by a word is enough (e.g., for items dealing with the sociative case of Basque nouns). In contrast, there are some topics that require chunk information (e.g., when working with determiners).

There is no predefined list to limit the possible values of the *topic* attribute. However, we have limited the *type* attribute to an enumerated list, as the question types in general are limited and definable. The predefined question types are: FBQ, error correction, MCQ, word formation and short answer. Although it is possible to foresee more question types, such as true/false and matching, we have not added them because we have not implemented them. However, in an XML schema, enumeration facilitates extension, and so there is no problem in adding a new value.

Apart from being able to represent different types of question, the model also represents items in several languages. In order to distinguish between the items, the **question** element contains the *language* attribute. This model is not intended to represent test items with no text in their foundations, as could happen in a purely mathematical scenario, and so the *language* attribute is compulsory. In addition, every item corresponds to at least one level or grade. In order to specify this characteristic, the **question** contains the *level* attribute. Thus far, we have included the six levels (A1, A2, B1, B2, C1, C2) defined in the European Framework of Reference for Languages and the grades of OSE (DBH1, DBH2, DBH3, DBH4), high school (BATX1, BATX2) and university (UNI) for domain-specific items. These grades have been defined taking into account the grade level of the Basque Country (cf., section II.3). When items belong to a grade, they also have the *subject* attribute, which is optional, and which specifies the domain or subject of the items. For instance, in our experiments, we have generated items to the “DBH2” *level* in the “science” *subject*.

III.3.1.2 Answer focus

As previously mentioned, the **answer_focus** contains the chunk of the sentence in which the topic appears. It also contains the additional information which is not part of the source sentence but that is necessary in order to deal with the topic. Finally, it also includes the components of the question types which are not part of the sentence, such as the distractors.

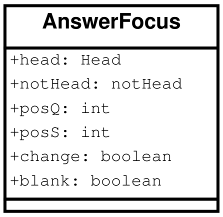


Figure III.3: The answer_focus element

A **question** can have more than one **answer_focus** dealing with the same topic, and they can be related or unrelated. For example, FBQs can have more than one blank and error correction items can comprise two errors, as shown in example III.3.1.

Example III.3.1 (Error correction — Two errors)

Urrak eta intzaurrak jan genituen sagardotegian.[10].

The given example shows two errors that students have to correct and which are represented by two different answer focuses. Both errors are orthographic errors,[11] but these errors are independent of one another. In contrast, an item can contain two answer focuses which are related to one another. For instance, an error correction question can contain two errors that refer to the same conceptual error, as displayed in example III.3.2.

Example III.3.2 (Error correction — Same conceptual error)

Nik mendi bat ikusten da.[12]

[10] Hurrak eta intxaurrak jan genituen sagardotegian (We ate hazeinuts and wafnuts in the cyder house.)

[11] We have not implemented this type of error.

[12] Nik mendi bat ikusten dut. (I see a mountain.)

In the given item, the aim is to work with the verb ikusi (to see) and its use. Thus, students need to know that in the given context, the verb ikusi has to appear with the DU auxiliary type (ikusten DUT) (cf., Appendix A), where the absolutive case works as the object of the sentence and the ergative functions as the subject of the sentence.

As regards chunks, the **answer_focus** consists of two elements: the **head** element which contains the necessary information within the chunk to address the topic and the **notHead** element, which comprises the rest of the words of the chunk.

We have already mentioned that, depending on the question type, the order of the chunks in the source sentence and the stem of the item can vary. In order to represent this variation, the model contains the *posS* and *posQ* attributes. The former indicates the original position of the chunk and the latter the position of the chunk in the item. It has been also noted that some languages such as Basque are considered (up to a point) to be free from word order. It is possible to represent the word order freedom of such languages by means of the *change* attribute, which limits which chunks can change the order when setting the final item. These three attributes are part of the chunk belonging to the **answer_focus** and of the rest of the chunks of the sentence. Example III.3.3 shows a short answer question which explains these attributes.

Example III.3.3 (Short answer example)

NOIZ egiten dute karroza-desfile ikusgarria?

WHEN is an amazing float parade held?

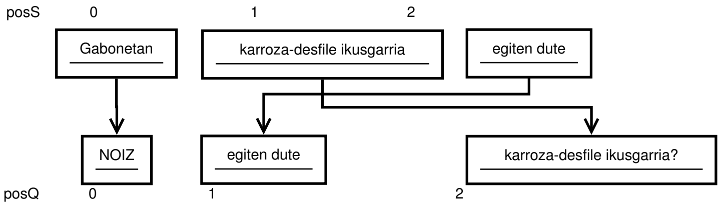


Figure III.4: Short answer question: order example

In example III.3.3, there is a difference between the original sentence Gabonetan karroza-desfile ikusgarria egiten dute (At Christmas an amazing float parade is held) and the question which has been generated in terms of chunk order. As Figure III.4 represents, what was originally 0 - 1 - 2 —the values of *posS*— has been transformed into a chunk order of 0 - 2 - 1 —the values of *posQ*—.

In addition, the *change* attribute establishes whether or not a chunk of the item can change the order. In the given example, it has a "false" value for posQ=0 and posQ=1 and a "true" for the rest. Apart from covering the word order freedom, offering this "changeability" information has two advantages for the assessment application. On the one hand, there is the option of producing questions which are superficially different but which are all the same. This is very useful for producing visually different tests to distribute to students. For instance, this is a way of preventing the option of copying. On the other hand, if an assessment application includes the automatic correction of students' responses, the values of the *change* attribute can establish all of the potential correct answers with different chunk order combinations.

Depending on the question type, the stem may include a blank or it may not. Among the implemented question types, in word formation and FBQs the **answer_focus** is always a blank, and MCQs can feature a blank. In contrast, error correction and short answer questions cannot have a blank within the stem. Therefore, the *blank* attribute of the **answer_focus** is used to define whether or not the **answer_focus** is a blank.

III.3.1.3 Head

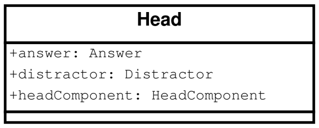


Figure III.5: The head element

As previously noted, the **head** element comprises the minimum list of words in the chunk which deal with the topic, as well as their linguistic information. In addition, it also contains new components which are not part of the original sentence, i.e., the **distractor** and the **headComponent** elements (see Figure III.5). In this way, the **head** contains three elements: the **answer** element, the **distractor** element and the **headComponent** element. The **answer** element is the only mandatory element, as it contains the information which is common to all of the question types.

Answer

The **answer** element (see Figure III.6) contains the minimum list of words in which the topic in question appears, together with their linguistic analysis. Every word within the item is represented by means of the **word** element, independent of its being part of the stem, the answer, the distractor or the headComponent. Therefore, the **word** element contains a word which is part of a chunk and its position in the chunk, represented by the *pos* attribute. The **analysis** element is used to represent the linguistic analysis of the corresponding **word** element. That is why it also features the *pos* attribute.

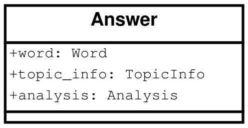


Figure III.6: The answer element

Example III.3.4 (MCQ example)

Hainbat ariketaren bidez .... askatu behar dugu.[13]
((We) have released .... by means of some exercises)
a) gure gorputzaren blokeoa (the stiffening of our bodies)
b) gure gorputzaren blokeoarekin (with the stiffening of our bodies)
c) gure gorputzaren blokeoan (in the stiffening of our bodies)
d) gure gorputzaren blokeoak (the stiffenings of our bodies)

The aim of the item presented in example III.3.4 is to work with the absolutive case of Basque nouns (see Appendix A). The source sentence comprises three chunks: [Hainbat ariketaren bidez][14] [gure gorputzaren blokeoa][15] [askatu behar dugu].[16] The chunk containing the topic is gure gorputzaren blokeoa, but the information required in order to deal with the topic is only in the last word of the chunk, blokeoa (the stiffening), as its analysis contains the information about the absolutive case. The model represents this distinction

[13] Some teachers could prefer to offer as candidate just the head of the answer focus. By means of an editing application the task of updating the item is an easy task so that it could be easily offered it.

[14] [by means of some exercises]

[15] [the stiffening of our bodies]

[16] [(we) have released]

by means of the **head** and **notHead** elements, and it presents the words together with their positions in the chunk as well as their linguistic information. This linguistic information is represented in the model as it is obtained from the analyser. The model offers this information as additional information, and that is why it is not represented in a standardised way. Within the XML representation, these elements would be represented as displayed in example III.3.5.

Example III.3.5 (Linguistic information example)

```
<head>
  <answer>
    <word pos="2"> blokeoa </word>
    ...
    <analysis pos="2">
      ("blokeo" IZE ARR DEK ABS NUMS MUGM @OBJ @SUBJ @PRED %SIB)
    </analysis>
  </answer>
  ...
</head>
<notHead>
  <word pos="0"> gure </word>
  <word pos="1"> gorputzaren </word>
  <analysis pos="0">
    ("gu" IOR PERARR NUMP GU DEK GEN @IZLG> @<IZLG %SIH)
  </analysis>
  <analysis pos="1">
    ("gorputz" IZE ARR DEK GEN NUMS MUGM @IZLG> @<IZLG)
  </analysis>
</notHead>
```

The **answer** element contains one last element which offers an overview of the information employed and generated in order to deal with the topic. This element is the **topic_info** element. This information is presented in a structured way and varies depending on the topic and the question type. Although the values of the *topic* attribute[17] are unlimited, ArikIturri is able to generate items to deal with declension cases, verb forms, determiners and vocabulary. Based on these general topics, we have defined the components of the **topic_info** element with the purpose of grouping the information which is related to them. Thus far, the **topic_info** element contains three elements: **linguistic_info**, the **lemma** and the **function**.

The **linguistic_info** and **lemma** elements represent information relating to the *topic* that stems from the linguistic features of the **answer**. In con-

[17] The *topic* attribute corresponds to the **question** element.

trast, the **function** element symbolises the information used to generate new components of an item relating to the answer.

Topic	**Value(s)**
Declension	case
	person
Verb	person
	mode
	tense
	paradigm
	subject
	object
	indirect object
Determiner	erroneous_phrase
Vocabulary	category
	case

Table III.2: All possible values of the **linguistic_info** element based on the *topic*.

The **linguistic_info** element has been defined in order to represent the linguistic information required in order to deal with the *topic*. Although it is a general element, the information offered is language-dependent. Therefore, depending on the *topic*, the information provided by this element varies. Table III.2 summarises all these values.

When the *topic* of an item has the declension value, ArikIturri requires the information relating to the case and person in order to deal with it. In example III.3.4, we have mentioned that the word blokeoa (the stiffening) contains the necessary information to deal with the topic. Looking at its analysis, the system extracts the useful information to present to the experts, as example III.3.6 shows.

Example III.3.6 (Declension — Linguistic information)

```
<head>
  <answer>
    <word pos="2"> blokeoa </word>
    <topic_info>
      <linguistic_info>
      case(absolutive);person(singular)
      </linguistic_info>
    </topic_info>
    <analysis pos="2">
      ("blokeo" IZE ARR DEK ABS NUMS MUGM @OBJ @SUBJ @PRED %SIB)
    </analysis>
  </answer>
   ...
</head>
```

In this way, experts can consult the linguistic information regarding the key of the item that has been used in the generation process. Although the given information contains linguistic terminology, this information is consulted by users, i.e., teachers who do not have problems understanding the information.

With regards to verbs, the system consults the paradigm, mode and tense values extracted from the analysed verb form. For instance, if an item works with the correct use of the DU paradigm and with the `dut` key, the **linguistic_info** element shows the information presented in example III.3.7.

Example III.3.7 (Verb — Linguistic information)

```
<head>
  <answer>
  <word pos="0"> dut </word>
  <topic_info>
    <linguistic_info>
    mode(indicative);tense(present);subj(it);obj(it)
    </linguistic_info>
  </topic_info>
  <analysis pos="0">
    "ukan" ADT PNT A1 NOR_NORK NR_HURA NK_HARK w10,L-A-ADT-4,lsfi17 @-↩
        JADNAG %ADIKAT
  </analysis>
  </answer>
</head>
```

In example III.3.7, the **verb** element offers information regarding the mode, tense and persons, as this information is used to deal with the DU verb paradigm. The way to represent this type of information is not established by the schema, and so the generator itself shows the information in an easy-to-understand way.

In the experiments presented in sections VI.3 and VI.4, when dealing with vocabulary, the only information required as regards the key to dealing with the *topic* is the category and case. Note that while section VI.4 deals with the Basque language, section VI.3 proposes items as part of an English language learning scenario.

Finally, as will be explained in section V.3.2, in our experiments, the input source for dealing with determiners is an erroneous determiner phrase. This is why the linguistic information required comprises the entire erroneous phrase, which is represented by means of the **erroneous_phrase** element.[18] For instance, with the source determiner error phrase *Mendia bat (*a one mountain), the **linguistic_info** element would contain the following information.

Example III.3.8 (Determiner — Linguistic information)

```
<head>
  <answer>
   ...
  <topic_info>
    <linguistic_info>
    <erroneous_phrase> noun + det + art <erroneous_phrase>
    </linguistic_info>
  </topic_info>
   ...
  </answer>
</head>
```

However, the key offered in the **answer** element by means of the **word** element always corresponds to a correct form. This is why, in such cases, ArikIturri automatically generates the correct answer (cf., section V.3.2). When this happens, the **topic_info** element contains the **artificial** tag to represent this action.

Thus, the **artificial** element is added when the key of an item is not part of the source sentence and is artificially created. From the users' point of view, it can also be interesting to see how this answer has been generated.

[18] In cases in which the starting point is a correct determiner, the entire phrase would also be offered. As previously mentioned, the marking to deal with this topic is carried out at the phrase level.

That is why the model contains not only the **artificial** element, but also the **function** element. Therefore, the **function** element expresses the way in which the key of the item has been generated. Thus far, we have detected two different situations. On the one hand, there are situations in which the source sentence does not contain the correct form of the key. On the other hand, there are cases in which it would be of interest to substitute the key of the source sentence. Table III.3 presents some of the values this element can take.

Value
synonym
correct_form
change_verb_form

Table III.3: The function element.

The correct_form value is used when the key is automatically corrected. The rest of the values (synonym and change_verb_form) are related to the replacement of the key. The latter set of values can be used to change the difficulty of the item, to study the influence of the key in a text (cf., section VI.4.3.4) or even to create new items. Let us imagine that the aim of ArikIturri is to generate an item, the topic of which is the verb based on the following source sentence:

Liburua erosi du. ((Someone) has bought the book.)

The *answer focus identifier* module detects that the verb erosi du (has bought) is of the DU auxiliary type (see Appendix A), which means that the ergative case is the subject and the absolutive is the direct object of the clause. The object is Liburua (the book) and the subject, the third-person singular, is elided. As a consequence, it is possible to change the verb form as follows:

Liburua erosi dut. ((I) have bought the book.)

Therefore, it is possible to replace the verb form by substituting the subject (elided) and to still have an item which will work with the DU paradigm. This change is reflected in the instance of the model, as example III.3.9 shows.

Example III.3.9 (Verb form — Replacement)

```
<head>
  <answer>
    <word pos="1"> dut </word>
    <topic_info>
      <linguistic_info>
      mode(indicative);tense(present);subj(I);obj(it)
      </linguistic_info>
      <function>
      change_verb_form(subj,(s)he)
      </function>
      <artificial>1</artificial>
    </topic_info>
    <analysis pos="1">
      "*edun" ADL A1 NOR_NORK NR_HURA NK_NIK @+JADLAG %ADIKATBU
    </analysis>
  </answer>
   ...
</head>
<notHead>
  <word pos="0"> erosi </word>
  <analysis pos="0">
    "erosi" ADI SIN PART BURU NOTDEK @-JADNAG %ADIKATHAS
  </analysis>
</notHead>
```

Therefore, the **function** element shows that there has been a verb form transformation in order to create a new answer.

The last component of the **topic_info** element is the **lemma** element, which shows the lemma of the answer when the *topic* or the question type requires. The lemma is provided when particular combinations of topic and question type occur: declension and verb topics for word formation and short answer question types. The word formation and short answer question types can offer the lemma of the key to answer the items, but in our experiments, this makes sense when dealing with grammar, that is, declension and verbs. Nonetheless, there is no restriction against allowing information relating to the lemma to be given for other topics such as determiners, for instance.

Let us view the concepts explained above within a real example.

Example III.3.10 (Short answer — Lemma)

NOIZ egiten dute karroza-desfile ikusgarria? (Gabon)
WHEN is an amazing float parade held? (Christmas)

The aim is to work on declension, and for this purpose, a short answer question is generated. The **topic_info** element presents the following information as regards the **lemma** element:

Example III.3.11 (Short answer — Lemma)

```
<head>
  <answer>
    <word pos="0">Gabonetan</word>
  <topic_info>
     ....
    <lemma> Gabon </lemma>
     ....
  </topic_info>
   ....
</head>
```

Therefore, the **lemma** element shows the lemma of the key to be used as a clue because when dealing with declension, it is more important to be able to create the corresponding inflected word than to guess the term. In contrast, we believe that there are cases in which the option of offering the **lemma** would give too much information to students.

The possible values of the **topic_info** element aim to be as general as possible in order to include all possible topics. However, it is not possible to guarantee complete coverage. Nonetheless, within an XML schema, there is the chance to add new elements by means of the **any** element which has been included in the **topic_info** element.

Apart from creating new correct answers, there are cases in which the system has to generate new components in order to build an item. When this occurs, the non-mandatory **distractor** or **headComponent** elements are part of the representation of the item. That is to say, the **head** element needs the **distractor** or **headComponent** element as well as the **answer** element in order to represent the information. These non-mandatory elements collect the components of the items which do not correspond to the source sentence and, depending on the question type, different information is generated.

Distractor

In the case of the MCQs and error correction questions, the system generates distractors which are represented by means of the **distractor** element.

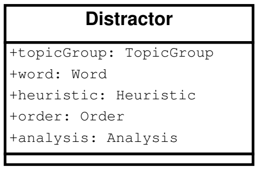


Figure III.7: The distractor element

From a linguistic point of view, a distractor is a list of words which are incorrect in the given context. This is why they are always linked to an **answer_focus**. In our approach, this list of words is generated automatically and the information used in this process is described by means of the **heuristic** and **order** elements (see Figure III.7). In addition, the **distractor** element contains the **topicGroup** component. Example III.3.12 shows the part of the XML code that represents the information regarding the require **distractor** instance.

Example III.3.12 (Distractor element — Example)

```
<distractor>
  <word pos="0">require</word>
  <topicGroup> B2 </topicGroup>
  <topicGroup> C1 </topicGroup>
  <heuristic>
    <type>similarity</type>
    <function>similarity_measure</function>
    <input> involve </input>
    <output> require </output>
  </heuristic>
  <order order="3">
    <method>corpus_based(ir_measure)</method>
    <value>0.30291</value>
  </order>
  <analysis pos="0">
      <lemma>require</lemma>
      <morphology>V PRES</morphology>
      <syntax>@+FMAINV %VA</syntax>
  </analysis>
</distractor>
```

As will be explained in section VI.3, this XML code corresponds to the distractor require that (with the the key involve) has been generated to work with the AWL in a given context. More specifically, the item has been generated to work with the vocabulary of the AWL. Thus, the heuristic to generate this distractor is related to the similarity concept.

In general, distractors that are generated by the system can be used to deal with the same topic at different language or grade levels. To specify this feature, the **distractor** element contains the **topicGroup** element. As will be explained in section VI.3, items focusing on the AWL are used in real scenarios at the B2 and C1 levels. The AWL is divided into mutually exclusive sub-lists, and some teachers use the first five sub-lists at the B2 level and the rest at the C1 level. In addition, at the C1 level, all of the words studied previously must be known. Therefore, the verb require, which is the distractor in the example above and part of the first sub-list, the distractor and consequently the item could be used at B2 and C1 levels.

As previously mentioned, the **heuristic** and **order** elements represent the information regarding the generation process of the distractor. Figure III.8 shows the components of both elements. Thus, the **heuristic** element first specifies the general type by means of the **type** element and then presents the function used to create the distractor in more detail using the **function** element. In addition, the **input** and **output** elements are used to represent the input and output of the function.

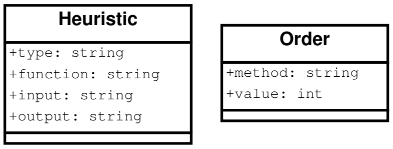


Figure III.8: The heuristic and order elements

As regards example III.3.12, the **heuristic** element specifies that the applied heuristic is related to the similarity concept (**type**). It also determines that, for the given input key involve, ArikIturri obtains the candidate distractor require by applying a similarity_measure.

Table III.4 shows all of the possible values of the **type** and **function** elements which have been used in different experiments[19] and are explained in more detail in sections V.3.1, V.3.2, VI.3 and VI.4.

When the same function creates more than one distractor, or when a preference is required, the **order** element is presented. The value of the *order* attribute specifies the order in which the distractors have been created, the lower the value, the higher the priority. The **method** element expresses the

[19] Notice that the replacement heuristic is used with two different topics: declension and determiners.

Type	Function
declension	change_finiteness
	replacement
verb	change_person
determiner	change_number
	replacement
	change_article
similarity	similarity_measure

Table III.4: Values of the **type** and **function** elements.

method used to attain this order and finally, the **value** element indicates the value obtained for the specified position.

In the case of the distractor displayed in example III.3.12, the same function (similarity_measure) was used to generate the rest of the distractors. Therefore, in addition to having the **heuristic** element, it also has the **order** element. The **order** element information shows that the distractor was the third in the generation process (order="3"). It also shows that in order to obtain this order, the system used a corpus-based information radius measure, obtaining a similarity value of 0.30291 (cf., section VI.3). Table III.5 presents all of the defined similarity measures to be used in sections VI.3 and VI.4.

Function	Method
similarity_measure	corpus_based(method)
	graph_based(method)
	combination(method1, method2)

Table III.5: Values of the **method** element based on the **function** element.

Although almost all of the implemented heuristics presented in chapters V and VI work with a single word, with the exception of determiner tests, the **distractor** element has been defined as a list of words which are incorrect in the given context. That is to say, the model has been designed to be able to represent items in which the distractors are created at chunk level. In fact, it is easy to imagine this type of distractor (see example III.3.13).

Example III.3.13 (MCQ — Distractors at chunk level)

Bihar liburua ....
a) erosi dut *b) erosiko dut*
c) erosiko ditut *c) erosi nau*

Example III.3.13 presents an hypothetical item which addresses the verb with the source sentence Bihar liburua erosiko dut (I will buy the book tomorrow). In order to generate the distractors, the system applies different heuristics which could be expressed as they are in example III.3.14.[20]

Example III.3.14 (Distractors at chunk level)

```
<distractor>
  <word pos="0">erosi</word>
  <word pos="1">dut</word>
  <topicGroup> B1 </topicGroup>
  <heuristic>
    <type>verb</type>
    <function>change_verb</function>
    <input> erosiko </input>
    <output> erosi </output>
  </heuristic>
   ....
</distractor>
<distractor>
  <word pos="0">erosiko</word>
  <word pos="1">ditut</word>
  <topicGroup> B1 </topicGroup>
  <heuristic>
    <type>verb</type>
    <function>change_object</function>
    <input> dut </input>
    <output> ditut </output>
  </heuristic>
   ....
</distractor>
<distractor>
  <word pos="0">erosi</word>
  <word pos="1">nau</word>
  <topicGroup> B1 </topicGroup>
  <heuristic>
    <type>verb</type>
    <function>change_verb_and_object</function>
    <input> erosiko dut </input>
    <output> erosi nau </output>
  </heuristic>
   ....
</distractor>
```

[20] We have not implemented this type of heuristics.

The “change_verb” and “change_object” values of the **function** element would work with a single word, but the “change_verb_and_object” function would take as input the verb phrase erosiko dut and would change features of both words.[21]

HeadComponent

The last component of the **head** element is the **headComponent**. This element collects specific information relating to the question type which is not part of the answer nor the distractor, but which is related to the answer focus. In our experiments with interrogative statements (cf., section V.2), the **headComponent** was part of an instance. As the source sentence was an affirmative statement, ArikIturri had to generate an interrogative statement in order to deal with the topic (cf., section V.2). Thus, in the given case, it was necessary to generate at least the corresponding wh-word. This contains the *type* attribute to specify the element type and the **rule** element (see Figure III.9) to show ways of generating this new component.

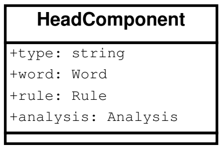


Figure III.9: The headComponent element

The defined **headComponent** elements are focused on the generation of wh-words, but there are some variations: sometimes the **headComponent** is a single word and at other times it is a list of words. Example III.3.15 shows a short answer question which deals with declension.

Example III.3.15 (Short answer — HeadComponent)

NOIZ egiten dute karroza-desfile ikusgarria? (Gabon)

WHEN is an amazing float parade held? (Christmas)

The **headComponent** element is represented in the model as shown in example III.3.16.

[21] These functions are not represented in our model.

Example III.3.16 (Short answer — HeadComponent — XML)

```
<headComponent type="wh_word">
  <word pos="0">NOIZ</word>
  <rule>wh_word(answer)</rule>
  <analysis pos="0">
       ("noiz" ADB GAL)
  </analysis>
</headComponent>
```

Therefore, this XML code corresponds to the wh-word and the rule expresses that by applying wh_word("Gabonetan") the system will obtain the wh-word NOIZ (WHEN).

However, the interrogative component can comprise a list of words, that is, a wh-word followed by at least one word. For instance, from the source sentence Baserritar bakoitzak bost euro ordaindu behar izan ditu (Each farmer has paid five euro), a short answer question to test the student's memory can be generated in different ways.

Example III.3.17 (Short answer example)

ZENBAT EURO ordaindu behar izan ditu baserritar bakoitzak?
HOW MANY EURO has each farmer paid?

The interrogative component of the short answer question in example III.3.17 is ZENBAT EURO (HOW MANY EURO). Thus, once the term to be tested is detected by ArikIturri, the corresponding wh-word is generated and the rest of the elements of the source sentence are kept. However, it is also possible to generate another short answer question (see example III.3.18) which is also related to the same term.

Example III.3.18 (Short answer example)

ZENBAT MONETA ordaindu behar izan ditu baserritar bakoitzak?
HOW MUCH MONEY has each farmer paid?

In example III.3.18, the interrogative component is ZENBAT MONETA (HOW MUCH MONEY). Both examples show the **headComponent** elements with a list of words. However, this second item has a new characteristic. When tasks involve memorising or understanding, it is common to change some components of the source sentence to make the question harder.

If this happens, the **rule** element not only expresses the way of creating the wh-word, but also the way of generating the new word, as example III.3.19 shows.

Example III.3.19 (Short answer example)

```
<headComponent type="wh_word">
  <word pos="0">ZENBAT</word>
  <word pos="1">moneta</word>
  <rule>wh_word(answer)</rule>
  <rule>hyperonym(euro)</rule>
</headComponent>
```

Therefore, in example III.3.19, the system substitutes the original word euro (euro) with the hyperonym moneta (money).

Finally, it is also possible to generate a question relating to the same term using only a wh-word (see example III.3.20).

Example III.3.20 (Short answer example)

ZENBAT ordaindu behar izan du baserritar bakoitzak?
HOW MUCH has each farmer paid?

In order to generate the short answer question presented in example III.3.20, apart from the corresponding wh-word, it is necessary to change the verb form in order to generate a correct interrogative statement. When this happens, this verb is considered to be a new **headComponent** because it is related to the **answer focus**.

Context

The last component of a **question** is the **context** element. As previously explained, this represents the chunks of the sentence which are not part of the answer focus of an item. Thus, as Figure III.10 shows, while the **context** element contains only a list of chunks, the **chunk** element represents all of the particularities of each chunk. More specifically, each chunk is represented by means of a list of **word** elements, together with their corresponding linguistic **analysis**.

As has already been pointed out, the order of the chunks can vary between the source sentence and the stem of the item, as well as when defining the

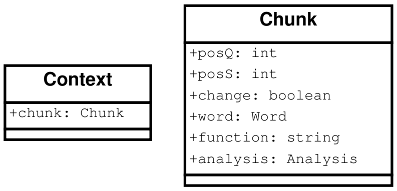


Figure III.10: The context and chunk elements

final item. That is why the **chunk** element adopts the *posS*, *posQ* and *change* attributes.[22]

One final characteristic of the **chunk** element is the option of changing the words by replacing some of them with new ones. Although we have not implemented it, the model offers the **function** component within the **chunk** element to express this replacement. This is an optional element that it is only used when a new word is created and is not related to the answer focus.

Finally, it is necessary to mention that there is no explicit way of representing a test as a whole. We believe that an external application should be responsible for this task. Indeed, ArikIturri is independent of any application which may use the items generated by it, and it is the application which determines the main characteristics of the questions to be generated and the way in which they are shown. The model offers all of the necessary information to generate tests with isolated questions, as well as with test items embedded in a text.

III.3.2 Examples

In the following, instead of showing the complete schema itself, we present one example for each type of question to show the way they are represented by the model. The detailed XML schema is available in Appendix B.

III.3.2.1 FBQs

FBQs have at least one blank that students need to fill in.

[22] The **answer_focus** element includes the same three attributes.

Example III.3.21 (FBQ example)

Barreari eustea .... batere ona ez osasun fisikorako ez mentalerako (Holding your laughter .... good at all for neither your physical nor your mental health.)

The corresponding XML representation is presented in example III.3.22.

Example III.3.22 (FBQ example — XML)

```
<question topic="verb" level="C1" source="t3_119.txt" pos="245" type="Fill-in-the-blank" language="eu">
  <answer_focus posQ="2" posS="2" change="false" blank="true">
    <head>
      <answer>
        <word pos="1">da</word>
        <topic_info>
          <linguistic_info>
          mode(indicative);tense(present);subj(it)
          </linguistic_info>
        </topic_info>
        <analysis pos="1">"izan" ADT PNT A1 NOR NR_HURA @+JADNAG %ADIKATBU</analysis>
      </answer>
    </head>
    <notHead>
      <word pos="0">ez</word>
      <analysis pos="0">"ez" PRT EGI @PRT %ADIKATHAS</analysis>
    </notHead>
  </answer_focus>
  <context>
    <chunk posQ="0" posS="0" change="false">
      <word pos="0"> Barreari </word>
      <analysis pos="0"> "barre" IZE ARR DAT NUMS MUGM HAS_MAI @ZOBJ %SINT </analysis>
    </chunk>
    <chunk posQ="1" posS="1" change="false">
      <word pos="1"> eustea </word>
      <analysis pos="1"> "eutsi" ADI SIN ADIZE KONPL ABS @-JADNAG_MP_OBJ %ADIKAT
    </chunk>
    <chunk posQ="3" posS="3" change="false">
      <word pos="0">batere</word>
      <word pos="1">ona</word>
      <analysis pos="0">"batere" DET DZG MG ZERO @ID> %SIH</analysis>
      <analysis pos="1">"on" IZE ARR BIZ- ABS NUMS MUGM @OBJ %SIB</analysis>
    </chunk>
    <chunk posQ="4" posS="4" change="false">
      <word pos="0">ez</word>
      <analysis pos="0">"ez" PRT EGI @PRT</analysis>
    </chunk>
    <chunk posQ="5" posS="5" change="false">
```

```
      <word pos="0">osasun</word>
      <word pos="1">fisikorako</word>
      <analysis pos="0">"osasun" IZE ARR BIZ- ZERO @KM> %SIH</analysis>
      <analysis pos="1">"fisiko" ADJ ARR IZAUR- ABS MG @ADLG %SIB</↩
          analysis>
    </chunk>
    <chunk posQ="6" posS="6" change="false">
      <word pos="0">ez</word>
      <analysis pos="0">"ez" PRT EGI @PRT</analysis>
    </chunk>
    <chunk posQ="7" posS="7" change="false">
      <word pos="0">mentalerako</word>
      <analysis pos="0">"mental" ADJ ARR IZAUR- ABS MG @ADLG</analysis>
    </chunk>
  </context>
</question>
```

The attributes of the *question* element have the following values:

```
<question topic="verb" level="C1" source="t3_119.txt" pos="245" type="Fill↩
    -in-the-blank" language="eu">
```

These values correspond to an item designed to deal with Basque verbs at the C1 language level. In addition, it presents the source of the sentence (t3_119.txt) and its corresponding position in it (245). As with every question represented by our model, the element contains two elements: the **answer_focus** and the **context**. The item's source sentence Barreari eustea ez da batere ona ez osasun fisikorako ez mentalerako. (Holding your laughter is not good at all for neither your physical nor your mental health.) has eight chunks which are divided into the answer_focus and context elements. The chunks are: Barreari, eustea, ez da, batere ona, ez, osasun fisikorako, ez and mentalerako.[23] As the topic is verbs, the answer_focus contains the verb phrase ez da while the rest of the chunks are part of the context element. For each chunk, the model shows the words together with their linguistic analysis.

```
<chunk posQ="0" posS="0" change="false">
  <word pos="0"> Barreari </word>
  <analysis pos="0"> "barre" IZE ARR DAT NUMS MUGM HAS_MAI @ZOBJ %SINT </↩
      analysis>
</chunk>
```

[23] your laughter, holding, is not, good at all, neither, for your physical health, nor, for your mental health

The chunk is composed of a single word (Barreari) which corresponds to a NP marked with the dative case. This is the first chunk of the source sentence (posS="0") and also of the item (posQ="0") which cannot change its order in the item (change="false").

Regarding the answer_focus, there is one single answer_focus which corresponds to the verb in question.

```
<answer_focus posQ="2" posS="2" change="false" blank="true">
  <head>
    <answer>
      <word pos="1">da</word>
      <topic_info>
        <linguistic_info>
          mode(indicative);tense(present);subj(it)
        </linguistic_info>
      </topic_info>
      <analysis pos="1">"izan" ADT PNT A1 NOR NR_HURA @+JADNAG %ADIKATBU</↩
          analysis>
    </answer>
  </head>
  <notHead>
    <word pos="0">ez</word>
    <analysis pos="0">"ez" PRT EGI @PRT %ADIKATHAS</analysis>
  </notHead>
</answer_focus>
```

As with FBQs, there is a blank that needs to be filled in, the blank attribute of the answer_focus element has the "true" value. The verb phrase that is part of the answer focus is composed of two words, but only one is necessary in order to deal with the topic. This is why the element has the **head** and **notHead** elements. While the head element contains all of the necessary information to deal with the topic, the notHead element comprises the rest of the words of the chunks as well as their corresponding linguistic analysis. In the given example, the word ez (not) is used to build a negative sentence. To build negative sentences in Basque, the negation word ez (not) precedes the inflected verb, meaning that it is part of the verb phrase but that it is not indispensable for dealing with the topic of the item. In addition to the inflected verb da (is) as part of the answer element, the model represents the linguistic information extracted from the chunk to deal with the topic by means of the topic_info element and, more specifically, the **linguistic_info** element. As the topic is verbs and a DA paradigm, this element offers information regarding the mode (indicative), the tense (present) and the persons (it) that correspond to the subject of the verb.

Thanks to the given XML code, we have described all of the attributes and elements of the schema that are part of FBQs. In the following sections, we will only explain new tags and attributes corresponding to each question type.

III.3.2.2 Word formation

As regards the representation of FBQs and word formation items, the main difference between them is that in word formation items, there is a word relating to the blank that must be changed.

Example III.3.23 (Word formation example)

Barreari eustea ez .... (izan) batere ona ez osasun fisikorako ez mentalerako
(Holding your laughter .... (to be) not good at all for neither your physical nor your mental health.)

The corresponding XML representation is shown in example III.3.24.

Example III.3.24 (Word formation example — XML)

```
<question topic="verb" level="C1" source="t3_119.txt" pos="245" type="↩
    word formation" language="eu">
  <answer_focus posQ="2" posS="2" change="false" blank="true">
      <head>
        <answer>
          <word pos="1">da</word>
          <topic_info>
            <linguistic_info>
            mode(indicative);tense(present);subj(it)
            </linguistic_info>
            <lemma>izan</lemma>
          </topic_info>
          <analysis pos="1">"izan" ADT PNT A1 NOR NR_HURA @+JADNAG %↩
              ADIKATBU</analysis>
        </answer>
      </head>
      <notHead>
        <word pos="0">ez</word>
        <analysis pos="0">"ez" PRT EGI @PRT %ADIKATHAS</analysis>
      </notHead>
  </answer_focus>
  <context>
    <chunk posQ="0" posS="0" change="false">
      <word pos="0"> Barreari </word>
      <analysis pos="0"> "barre" IZE ARR DAT NUMS MUGM HAS_MAI @ZOBJ %↩
          SINT </analysis>
    </chunk>
    <chunk posQ="1" posS="1" change="false">
      <word pos="1"> eustea </word>
      <analysis pos="1"> "eutsi" ADI SIN ADIZE KONPL ABS @-JADNAG_MP_OBJ↩
          %ADIKAT
    </chunk>
    <chunk posQ="3" posS="3" change="false">
      <word pos="0">batere</word>
      <word pos="1">ona</word>
      <analysis pos="0">"batere" DET DZG MG ZERO @ID> %SIH</analysis>
      <analysis pos="1">"on" IZE ARR BIZ- ABS NUMS MUGM @OBJ %SIB</↩
          analysis>
    </chunk>
    <chunk posQ="4" posS="4" change="false">
      <word pos="0">ez</word>
      <analysis pos="0">"ez" PRT EGI @PRT</analysis>
    </chunk>
    <chunk posQ="5" posS="5" change="false">
      <word pos="0">osasun</word>
      <word pos="1">fisikorako</word>
      <analysis pos="0">"osasun" IZE ARR BIZ- ZERO @KM> %SIH</analysis>
      <analysis pos="1">"fisiko" ADJ ARR IZAUR- ABS MG @ADLG %SIB</↩
          analysis>
    </chunk>
    <chunk posQ="6" posS="6" change="false">
      <word pos="0">ez</word>
      <analysis pos="0">"ez" PRT EGI @PRT</analysis>
    </chunk>
    <chunk posQ="7" posS="7" change="false">
      <word pos="0">mentalerako</word>
      <analysis pos="0">"mental" ADJ ARR IZAUR- ABS MG @ADLG</analysis>
    </chunk>
  </context>
</question>
```

As they have the same source sentence, the new element in the word formation item is the **lemma** element within the topic_info:

```
<topic_info>
  <linguistic_info>
  mode(indicative);tense(present);subj(it)
  </linguistic_info>
  <lemma>izan</lemma>
</topic_info>
```

This element is used to show the extra information that is going to be offered in order to create the correct word formation, that is, the lemma of the correct answer.

III.3.2.3 Error correction

Example III.3.25 shows a marked error correction item designed to deal with declension.

Example III.3.25 (Error correction example)

Hainbat ariketaren bidez gure gorputzaren blokeoarekin askatu behar dugu.

(We) have released with the stiffening of our bodies by means of some exercises.

The XML representation that corresponds to this item is shown in example III.3.26.

Example III.3.26 (Error correction example — XML)

```
<question topic="declension" level="C1" source="t3_119.txt" pos="46" ↩
    type="error-correction" language="eu">
  <answer_focus posQ="1" posS="1" change="true" blank="true">
    <head>
      <answer>
        <word pos="2">blokeoa</word>
        <topic_info>
          <linguistic_info>
          case(absolutive); person(singular)
          <linguistic_info>
        <topic_info>
        <analysis pos="2">
          ("blokeo"  IZE ARR DEK ABS NUMS MUGM @OBJ @SUBJ @PRED %SIB)
        </analysis>
      </answer>
      <distractor>
        <topicGroup> B2 </topicGroup>
        <topicGroup> C1 </topicGroup>
        <word pos="2">blokeoarekin</word>
        <heuristic>
          <type> declension </type>
          <function> replacement(basque, abs, soz) </function>
          <input> blokeoa </input>
          <output> blokeoarekin </output>
        </heuristic>
        <analysis pos="2">
        ("blokeo"  IZE ARR DEK SOZ NUMS MUGM @ADLG %SIB)
        </analysis>
      </distractor>
    </head>
    <notHead>
      <word pos="0">gure</word>
      <word pos="1">gorputzaren</word>
      <analysis pos="0">("gu"  IOR PERARR NUMP GU DEK GEN )</analysis>
      <analysis pos="1">("gorputz"  IZE ARR DEK GEN NUMS MUGM )</↩
          analysis>
    </notHead>
  </answer_focus>
  <context>
    <chunk posQ="0" posS="0" change="true">
      <word pos="0">Hainbat</word>
      <word pos="1">ariketaren</word>
      <word pos="2">bidez</word>
      <analysis pos="0">("hainbat"  DET DZG MG @ID> %SIH) </analysis>
      <analysis pos="1">("ariketa"  IZE ARR DEK GEN MG )</analysis>
      <analysis pos="2">("bide"  IZE ARR DEK INS MG @ADLG %SIB)</↩
          analysis>
    </chunk>
     ...
  </context>
</question>
```

Thus, the new element in this type of item in comparison with FBQs and word formation questions is the **distractor** element. As previously mentioned, the distractor is an element of the item which is not part of the source sentence. ArikIturri generates this new element automatically and the model offers the information relating to this process within the **distractor** element.

```
<distractor>
  <topicGroup> B2 </topicGroup>
  <topicGroup> C1 </topicGroup>
  <word pos="2">blokeoarekin</word>
  <heuristic>
    <type> declension </type>
    <function> replacement(basque, abs, soz) </function>
    <input> blokeoa </input>
    <output> blokeoarekin </output>
  </heuristic>
  <analysis pos="2">
    ("blokeo"  IZE ARR DEK SOZ NUMS MUGM @ADLG %SIB)
  </analysis>
</distractor>
```

Thus, the distractor blokeoarekin (with the stiffening) is a plausible candidate in the B2 and C1 language levels (**topicGroup**). In addition, the **distractor** element shows that the applied heuristic is of the declension type that, by means of the replacement function, has transformed the key blokeoa (input) into the candidate distractor blokeoarekin (output). In other words, the heuristic has replaced the absolutive mark of the key with the sociative in order to generate the distractor.

III.3.2.4 MCQs

The representation of an error correction question type and an MCQ is basically the same. The only difference is that for MCQs, there is the option of generating more than one distractor. Thus, it is possible to apply different heuristics in order to generate these distractors or to apply the same one in order to generate a list of candidate distractors. For the latter, the **distractor** element contains the **order** element. Given the source sentence Although the problem may not always be clearly articulated, these women are seeking alternative rituals, example III.3.28 presents the corresponding XML code for three out of the 10 generated distractors and example III.3.27 the corresponding MCQ.

Example III.3.27 (MCQ example)

Although the problem may not always be clearly articulated, these women .... alternative rituals.
a. are seeking *c. are securing*
b. are obtaining *d. are perceiving*

Example III.3.28 (MCQ exampe — XML)

```
<question topic="vocabulary" level="C1" source="/home/jibalari/sc01a4/↩
    BNC/A/AC/ACL.txt.xml.gz" pos="273" language="en">
  <answerfocus posQ="5" change="false" posS="5" blank="false">
    <head>
      <answer>
        <word pos="1">seeking</word>
        <topic_info>
          <linguistic_info>
          category(verb); case(present continuous)
          </linguistic_info>
        </topic_info>
        <analysis pos="1">
          <lemma>seek</lemma>
          <morphology>ING</morphology>
          <syntax>@-FMAINV %VA</syntax>
        </analysis>
      </answer>
      <distractor>
       <topicGroup> C1 </topicGroup>
       <word pos="1">obtaining</word>
       <heuristic>
         <type>similarity</type>
         <function>similarity_measure</function>
         <input>seek</input>
         <output>obtain</output>
```

```
     </heuristic>
     <order order="10">
       <method>corpus_based(ir_measure)</method>
       <value>0.37646</value>
     </order>
     <analysis>
       <lemma>obtain</lemma>
       <morphology>ING</morphology>
       <syntax>@-FMAINV %VA</syntax>
     </analysis>
    </distractor>
    <distractor>
     <topicGroup> C1 </topicGroup>
     <word pos="1">securing</word>
     <heuristic>
       <type>similarity</type>
       <function>similarity_measure</function>
       <input>seek</input>
       <output>secure</output>
     </heuristic>
     <order order="9">
       <method>corpus_based(ir_measure)</method>
       <value>0.43479</value>
     </order>
     <analysis>
       <lemma>secure</lemma>
       <morphology>ING</morphology>
       <syntax>@-FMAINV %VA</syntax>
     </analysis>
    </distractor>
    ....
    <distractor>
      <topicGroup> C1 </topicGroup>
      <word pos="1">perceiving</word>
      <heuristic>
       <type>similarity</type>
       <function>similarity_measure</function>
       <input>seek</input>
       <output>perceive</output>
      </heuristic>
      <order order="1">
       <method>corpus_based(ir_measure)</method>
       <value>0.60379</value>
      </order>
      <analysis pos="1">
        <lemma>perceive</lemma>
        <morphology>ING</morphology>
        <syntax>@-FMAINV %VA</syntax>
      </analysis>
    </distractor>
    </head>
    <notHead>
      <word pos="0">are</word>
      <analysis pos="0">
        <lemma>be</lemma>
        <morphology>V PRES</morphology>
        <syntax>@+FAUXV %AUX</syntax>
        </analysis>
  </answerfocus>
  <context>
    <chunk posQ=”0” change=”false” posS=”0”>
      <word pos=”0”>Although</word>
      <analysis pos=”0”>
        <lemma>although</lemma>
        <morphology>CS</morphology>
        <syntax>@CS %CS</syntax>
      </analysis>
    </chunk>
    <chunk posQ=”1” change=”false” posS=”1”>
      <word pos=”0”>the</word>
      <word pos=”1”>problem</word>
      <analysis pos=”0”>
        <lemma>the</lemma>
        <morphology>DET</morphology>
        <syntax>@DN> %>N</syntax>
      </analysis>
      <analysis pos=”1”>
        <lemma>problem</lemma>
        <morphology>N NOM SG</morphology>
        <syntax>@SUBJ %NH</syntax>
      </analysis>
    </chunk>
    <chunk posQ=”2” change=”false” posS=”2”>
      <word pos=”0”>may</word>
      <word pos=”1”>not</word>
      <word pos=”2”>always</word>
      <word pos=”3”>be</word>
      <word pos=”4”>clearly</word>
      <word pos=”5”>articulated</word>
      <analysis pos=”0”>
        <lemma>may</lemma>
        <morphology>V AUXMOD</morphology>
        <syntax>@+FAUXV %AUX</syntax>
      </analysis>
      ....
      <analysis pos=”5”>
        <lemma>articulate</lemma>
        <morphology>EN</morphology>
        <syntax>@-FMAINV %VP</syntax>
      </analysis>
    </chunk>
    <chunk posQ=”3” change=”false” posS=”3”>
      <word pos=”0”>,</word>
      <analysis pos=”0”>
        <lemma>,</lemma>
        <morphology>empty</morphology>
        <syntax>empty</syntax>
      </analysis>
    </chunk>
    <chunk posQ=”4” change=”false” posS=”4”>
      <word pos=”0”>these</word>
      <word pos=”1”>women</word>
      <analysis pos=”0”>
        <lemma>this</lemma>
        <morphology>DET DEM PL</morphology>
        <syntax>@DN> %>N</syntax>
      </analysis>
      <analysis pos="1">
        <lemma>woman</lemma>
        <morphology>N NOM PL</morphology>
        <syntax>@SUBJ %NH</syntax>
      </analysis>
    </chunk>
    <chunk posQ="6" change="false" posS="6">
      <word pos="0">alternative</word>
      <word pos="1">rituals</word>
      <analysis pos="0">
        <lemma>alternative</lemma>
        <morphology>A ABS</morphology>
        <syntax>@A> %>N</syntax>
      </analysis>
      <analysis pos="1">
        <lemma>ritual</lemma>
        <morphology>N NOM PL</morphology>
        <syntax>@OBJ %NH</syntax>
      </analysis>
    </chunk>
    <chunk posQ="7" change="false" posS="7">
      <word pos="0">.</word>
      <analysis pos="0">
        <lemma>.</lemma>
        <morphology>empty</morphology>
        <syntax>empty</syntax>
      </analysis>
    </chunk>
  </context>
</question>
```

III.3.2.5 Short answer

The final type of question implemented by ArikIturri is the short answer question. As previously mentioned, a short answer question is composed of a wh-word that corresponds to the key of the item. This type of question therefore always contains at least one **headComponent** element.

Given the source sentence Otsailaren 25a arte, nahi duenak iritzia emateko aukera du (Those who want to do so have the opportunity to express their views until February 25th) let us imagine that ArikIturri produces the short answer question presented in example III.3.29 to test students' comprehension of the text.

Example III.3.29 (Short-answer Example)

NOIZ ARTE du nahi duenak iritzia emateko aukera?

UNTIL WHEN do those who want to do so have the opportunity to express their views?

Example III.3.30 shows the corresponding XML code.

Example III.3.30 (Short answer example — XML)

```
<answer_focus posQ="0" change="false" posS="0" blank="false">
  <head>
    <answer>
     <word pos="0"> Otsailaren </word>
     <word pos="1"> 25a </word>
     <word pos="2"> arte </word>
     <topic_info>
       <linguistic_info>
       magnitude(data);case(alative)
       </linguistic_info>
     </topic_info>
     <analysis pos="0"> "otsail" IZE ARR BIZ- GEN NUMS MUGM ZERO HAS_MAI↩
          @IZLG> %SIH </analysis>
     <analysis pos="1"> "25" IZE ZKI ABS NUMS MUGM ZEN_DEK </analysis>
     <analysis pos="2"> "arte" IZE ARR BIZ- ABS MG @ADLG %SIB </analysis↩
          >
    </answer>
    <headComponent type="wh_word">
      <word pos="0">NOIZ</word>
      <word pos="1">ARTE</word>
      <rule>wh_word(answer)</rule>
      <analysis pos="0"> "noiz" ADB GAL ZERO HAS_MAI @ADLG %SIH </↩
          analysis>
      <analysis pos="1"> "arte" IZE ARR BIZ- ABS MG @ADLG %SIB </↩
          analysis>
    </headComponent>
  </head>
</answer_focus>
<context>
  <chunk posQ="1" change="false" posS="1">
    <word pos="0"> nahi duenak </word>
    <analysis pos="0"> "nahi_izan" ADI ADK PNT ABS NUMP MUGM A1 NR_HURA ↩
        NK_HARK HAUL_EDBL @+JADNAG_MP %ADIKAT </analysis>
  </chunk>
  <chunk posQ="2" change="false" posS="2">
    <word pos="0"> iritzia </word>
    <analysis pos="0"> "iritzi" IZE ARR BIZ- ABS NUMS MUGM @OBJ %SINT </↩
        analysis>
  </chunk>
  <chunk posQ="3" change="false" posS="3">
    <word pos="0"> emateko </word>
    <analysis pos="0"> "eman" ADI SIN ADIZE GEL ZERO @-JADNAG_MP_IZLG> <↩
        /analysis>
  </chunk>
  <chunk posQ="4" change="false" posS="4">
    <word pos="0"> aukera </word>
    <analysis pos="0"> "aukera" IZE ARR BIZ- ABS NUMS MUGM AORG @OBJ %↩
        SINT </analysis>
  </chunk>
  <chunk posQ="5" change="false" posS="5">
    <word pos="0"> du </word>
    <analysis pos="0"> "ukan" ADT PNT A1 NOR_NORK NR_HURA NK_HARK @+↩
        JADNAG %ADIKAT </analysis>
  </chunk>
</context>
```

This short answer example is one of the simplest. In fact, the only difference from the source sentence is related to the generation of the corresponding wh-word. The rest of the chunks are the same and keep the same position as they occupied in the source sentence.

As previously explained, once the first version of the model was designed, it was adapted so that the model could be offered as an extension of QTI. Section III.3.3 explains this process.

III.3.3 QTI extension

We have already mentioned that:

> The IMS QTI specification has been designed to support both interoperability and innovation through the provision of well-defined extension points. These extension points can be used to wrap specialized [sic] or proprietary data in ways that allows it to be used alongside items that can be represented directly (IMS Global Learning Consortium, accessed 2010).

In addition:

> The main purpose of the QTI specification is to define an information model and associated binding that can be used to represent and exchange assessment items. For the purposes of QTI, an item is a set of interactions (possibly empty) collected together with any supporting material and an optional set of rules for converting the candidate's response(s) into assessment outcomes (IMS Global Learning Consortium, accessed 2010).

Based on both premises, we first looked at the option of representing the question types implemented by ArikIturri in QTI. Then, we examined ways to propose an extension of QTI in order to represent the information which is not explicitly encoded in the QTI model due to the fact that the items are based on the exchange of information.

As is pointed out in the IMS Global Learning Consortium (accessed 2010):

> A test is a group of assessmentItems with an associated set of rules that determine which of the items the candidate sees, in what order, and in what way the candidate interacts with them. The rules describe the valid paths through the test, when responses are submitted for response processing and when (if at all) feedback is to be given.

Thus, the QTI specification defines different types of item and, for our purposes, we focused on the "Simple Items". These items contain just one point of interaction, as example III.3.31, extracted from the IMS QTI specification, shows.

Example III.3.31 (MCQ example — QTI)

```
<!-- This example adapted from the PET Handbook, copyright University of↩
     Cambridge ESOL Examinations -->
<assessmentItem xsi:schemaLocation="http://www.imsglobal.org/xsd/↩
    imsqti_v2p1 http://www.imsglobal.org/xsd/imsqti_v2p1.xsd" identifier↩
    ="choice" title="Unattended Luggage" adaptive="false" timeDependent=↩
    "false">
  <responseDeclaration identifier="RESPONSE" cardinality="single" ↩
      baseType="identifier">
    <correctResponse>
      <value>ChoiceA</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="↩
      integer">
    <defaultValue>
      <value>0</value>
    </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p>Look at the text in the picture.</p>
    <p>
      <img src="images/sign.png" alt="NEVER LEAVE LUGGAGE UNATTENDED"/>
    </p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" ↩
        maxChoices="1">
      <prompt>What does it say?</prompt>
      <simpleChoice identifier="ChoiceA">You must stay with your luggage↩
           at all times.</simpleChoice>
      <simpleChoice identifier="ChoiceB">Do not let someone else look ↩
           after your luggage.</simpleChoice>
      <simpleChoice identifier="ChoiceC">Remember your luggage when you ↩
           leave.</simpleChoice>
    </choiceInteraction>
  </itemBody>
  <responseProcessing template="http://www.imsglobal.org/question/↩
       qti_v2p1/rptemplates/match_correct"/>
</assessmentItem>
```

In brief, the *assessmentItem* comprises "the information that is presented to a candidate and information about how to score the item (IMS Global Learning Consortium, accessed 2010)." In the given example, the *responseDeclaration* variable is declared to store the *correctResponse* value, that is, the correct answer for the item that corresponds to the *itemBody*. The *outcomeDeclaration* variable, declared by an outcome declaration, is used to represent the numerical value of the students' overall performance. Finally, the *itemBody* contains the rest of the components of the item, such as "text, graphics, media objects, and interactions that describe the item's content and information about how it is structured (IMS Global Learning Consortium, accessed 2010)."

It is fairly obvious that this type of item does not offer the option of representing the information relating to the source sentence, topic and generation process. This is why an extension point is needed. Looking at the IMS QTI Version 2.1 specification, the extension points offered in the QTI model, which are expressed by means of the <xsd:extension> tag, are related to the *baseValue*, *value* and *weight* classes. As a consequence, something derived from this specification is necessary in order to offer the information provided by our model.

Within the *assessmentItem* class, we propose two modifications. On the one hand, we propose to define an attribute to represent information regarding the **topic**. On the other hand, we propose to define the way to represent the **head** of the items defined in our model. After studying the QTI specification, we decided to restrict our updates to those features and not to specify the chunk information obtained from the analyser.[24]

As the following XML schema shows, we modified the *assessmentItem* class by adding an attribute (topic) and one element (head) to the QTI specification. In addition, the QTI specification contains the *identifier* attribute

[24] In cases in which the chunk information is required, our own model should be used.

designed to define the item. Based on the fact that this attribute is a string and it is a mandatory field, the information relating to the source sentence is encoded and displayed in it. As regards the new components, the *topic* attribute is defined as optional and the **head** element has the option of not been instanced (minOccurs=“0”), thus, there is still the option of defining the items in the same way as in the QTI specification.

Listing III.1: Modified assessmentItem

```
<!-- Class: assessmentItem -->

<xsd:attributeGroup name="assessmentItem.AttrGroup">
  <xsd:attribute name="identifier" type="string.Type" use="required"/>
  <xsd:attribute name="title" type="string.Type" use="required"/>
  <xsd:attribute name="label" type="string256.Type" use="optional"/>
  <xsd:attribute ref="xml:lang"/>
  <xsd:attribute name="adaptive" type="boolean.Type" use="required"/>
  <xsd:attribute name="timeDependent" type="boolean.Type" use="required"/>
  <xsd:attribute name="toolName" type="string256.Type" use="optional"/>
  <xsd:attribute name="toolVersion" type="string256.Type" use="optional"/>
  <xsd:attribute name="topic" type="string.Type" use="optional"/>
</xsd:attributeGroup>

<xsd:group name="assessmentItem.ContentGroup">
  <xsd:sequence>
    <xsd:element ref="responseDeclaration" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="outcomeDeclaration" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="templateDeclaration" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="templateProcessing" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="stylesheet" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="itemBody" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="responseProcessing" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="modalFeedback" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="head" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:group>

<xsd:complexType name="assessmentItem.Type" mixed="false">
  <xsd:group ref="assessmentItem.ContentGroup"/>
  <xsd:attributeGroup ref="assessmentItem.AttrGroup"/>
</xsd:complexType>

<xsd:element name="assessmentItem" type="assessmentItem.Type"/>
```

The **head** element is based on the specification of our model. Thus, it contains information relating to the answer, the distractors and the headComponents. Although the chunk information is not offered in this specification, we decided to keep the linguistic information relating to the head in order to allow the direct importation of these elements from one model to the other.

Listing III.2: head Element

```
<xsd:complexType name=”head.Type”>
  <xsd:sequence>
    <xsd:element name=”answer” type=”answer.Type”/>
    <xsd:element name=”distractor” type=”distractor.Type” minOccurs=”0” ↩
        maxOccurs=”unbounded”/>
    <xsd:element name=”headComponent” type=”headComponent.Type” minOccurs=↩
        ”0” maxOccurs=”unbounded”/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name=”answer.Type”>
  <xsd:sequence>
    <xsd:element name=”word” type=”word.Type” maxOccurs=”unbounded”/>
    <xsd:element name=”topic_info” type=”topic_info.Type” />
    <xsd:element name=”analysis” type=”analysis.Type” maxOccurs=”unbounded↩
        ”/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name=”topic_info.Type”>
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name=”linguistic_info” type=”string.Type”/>
      <xsd:element name=”lemma” type=”string.Type”/>
    </xsd:choice>
    <xsd:element name=”function” type=”string.Type” minOccurs=”0”/>
    <xsd:attribute name=”artificial” type=”boolean.Type” use=”optional”/>
    <xsd:any minOccurs=”0”/>
  </xsd:sequence>
</xsd:element>

<xsd:complexType name=”distractor.Type”>
  <xsd:sequence>
    <xsd:element name=”topicGroup” type=”topicGroup.Type” use=”required” ↩
        minOccurs=”0” maxOccurs=”unbounded”/>
    <xsd:element name=”word” type=”word.Type” maxOccurs=”unbounded”/>
    <xsd:element name=”heuristic” type=”heuristic.Type”/>
    <xsd:element name=”order” type=”order.Type” minOccurs=”0”/>
    <xsd:element name=”analysis” type=”analysis.Type” maxOccurs=”unbounded↩
        ”/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name=”heuristic.Type”>
  <xsd:sequence>
    <xsd:element name=”type” type=”string.Type”/>
    <xsd:element name=”function” type=”string.Type”/>
    <xsd:element name=”input” type=”string.Type”/>
    <xsd:element name=”output” type=”string.Type”/>
  </xsd:sequence>
```

```
</xsd:complexType>

<xsd:complexType name="order.Type">
  <xsd:sequence>
    <xsd:element name="method" type="string.Type"/>
    <xsd:element name="function" type="integer.Type"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="headComponent.Type">
  <xsd:sequence>
    <xsd:element name="word" type="word.Type" maxOccurs="unbounded"/>
    <xsd:element name="rule" type="string.Type"/>
    <xsd:element name="analysis" type="analysis.Type" maxOccurs="unbounded↩
        "/>
  </xsd:sequence>
  <xsd:attribute name="type" type="string.Type" use="required"/>
</xsd:complexType>

<xsd:complexType name="word.Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="pos" type="integer.Type" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="analysis.Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="pos" type="integer.Type" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

In conclusion

In this chapter, we have presented the question model underlying ArikIturri. The model, which is flexible and general, describes the items generated by the system. The question model consists of several components; a question, conceptually, has three main components: the topic, the answer focus and the context. All the components have been extensively described in this chapter. Apart from our own model, an extension of QTI has also been proposed.

CHAPTER IV

Data analysis

In this chapter, we present the main resources used by ArikIturri. In addition to the main features of the resources, a study of the potential influence of the resources on the generated items is also presented. The way in which the automatically created items are evaluated is also expounded. This evaluation is based mainly on the item analysis theory.

IV.1 Study of the resources

This section groups together the main resources used during this dissertation. Their features as well as the tasks for which they have been used are explained. The aim of this analysis is also to study the influence of the resources used in the generation process, as the quality of those resources can determine the quality of the system and the generated questions.

The system makes use of various types of resource in order to generate items: corpora; ontologies; dictionaries; syntactic analysers and morphological generators. More specifically, the system mainly exploits the grammatical and semantic information contained therein, as our aim is to study how to make use of different linguistic information within different scenarios.

Corpora are one of the most commonly used resources, and are used in two different ways: as a source of questions and within the distractor generation task. We have worked with Basque and English corpora. When working with the Basque language, two general corpora (*Euskaldunon Egunkaria* newspaper and a Basque language learning corpus), a specialised corpus on science and technology (*ZT* corpus) and a Basque learners' corpus were used. When dealing with English, we used the British National Corpus (BNC), a mystery novel and the Web 1T 5-gram dataset. All of these corpora are explained within this section.

The dictionaries and ontologies were also used in the distractor generation task when generating items in the Basque language. As will be explained in this chapter, in some experiments, entries from two monolingual dictionaries and the Multilingual Central Repository (MCR) were consulted when producing the distractors.

The syntactic analysers are used mainly to complete the tagging of the input corpus at chunk level. Regarding the Basque language, ArikIturri uses *Ixati*, the analyser developed by the IXA research group. In the case of English texts, Connexor's Machinese Syntax was chosen. Both analysers were used at the very beginning of the generation process, but the linguistic information they offered was used in various other steps of the generation process.

In the following section, all of the resources used and their potential influence on the system are explained.

IV.1.1 NLP tools

IV.1.1.1 Basque analyser

The IXA research group has developed a robust cascaded syntactic analyser (Aduriz *et al.*, 2004) for linguistic analysis in Basque. The creation of this robust analyser was based on a shallow parser and implemented in sequential rule layers. The complete specification of the analyser can be found in Oronoz (2009).

Each layer of analysis uses as input the output of the previous layer and enriches the analysis with new linguistic information. The parsing process is based on finite state grammars which are encoded in the formats defined by Constraint Grammar (CG) and the Xerox Finite-State Tool (XFST). These two formalisms provide a useful methodology for dealing with free-order phrase components which occur in languages such as Basque. The modules and layers of analysis in the analysis chain can be seen in Figure IV.1.

In agglutinative languages like Basque, it is difficult to separate mor-

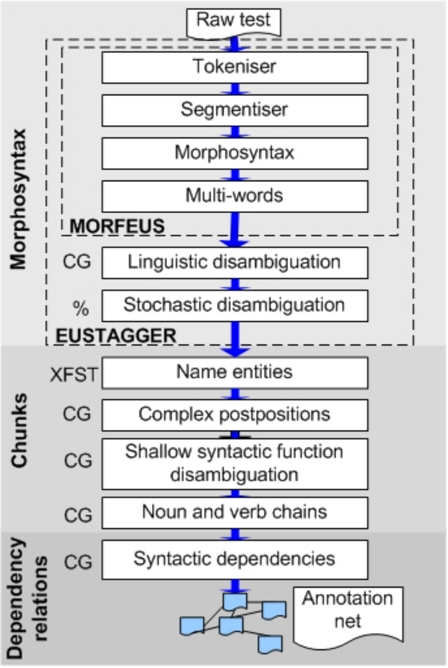


Figure IV.1: Analysis chain

phology from syntax, which is why we chose to consider morphosyntactic parsing for the first phase of the shallow syntactic analyser. The morphosyntactic analyser, *Morfeus* (Aduriz *et al.*, 1998), receives the input text and transforms it into tokens. For each token, it generates all of the possible combinations of lemmas and morphemes and proposes them together with the morphological information. The morphological analyser has a 99.3% level of correctness (Alegria *et al.*, 2003a).

For each word form contained within the output of *Morfeus*, *Eustagger* (Ezeiza, 2002), the lemmatiser and tagger for Basque, identifies the corresponding lemma and tag pair in the given context. This disambiguation process is carried out by means of linguistic rules and a stochastic method based on Markov models. It has a disambiguation precision level of 95.42%.

The chunker *Ixati* divides the source text into chunks. A chunk is composed of words which are syntactically related, and within the analysis chain, the group of words which are syntactically related are noun chains and verb chains. A noun chain can be an NP or derived from an NP (adjectives, adverbs, etc.). A verb chain is composed of the main verb of the clause and its corresponding co-occurrences.

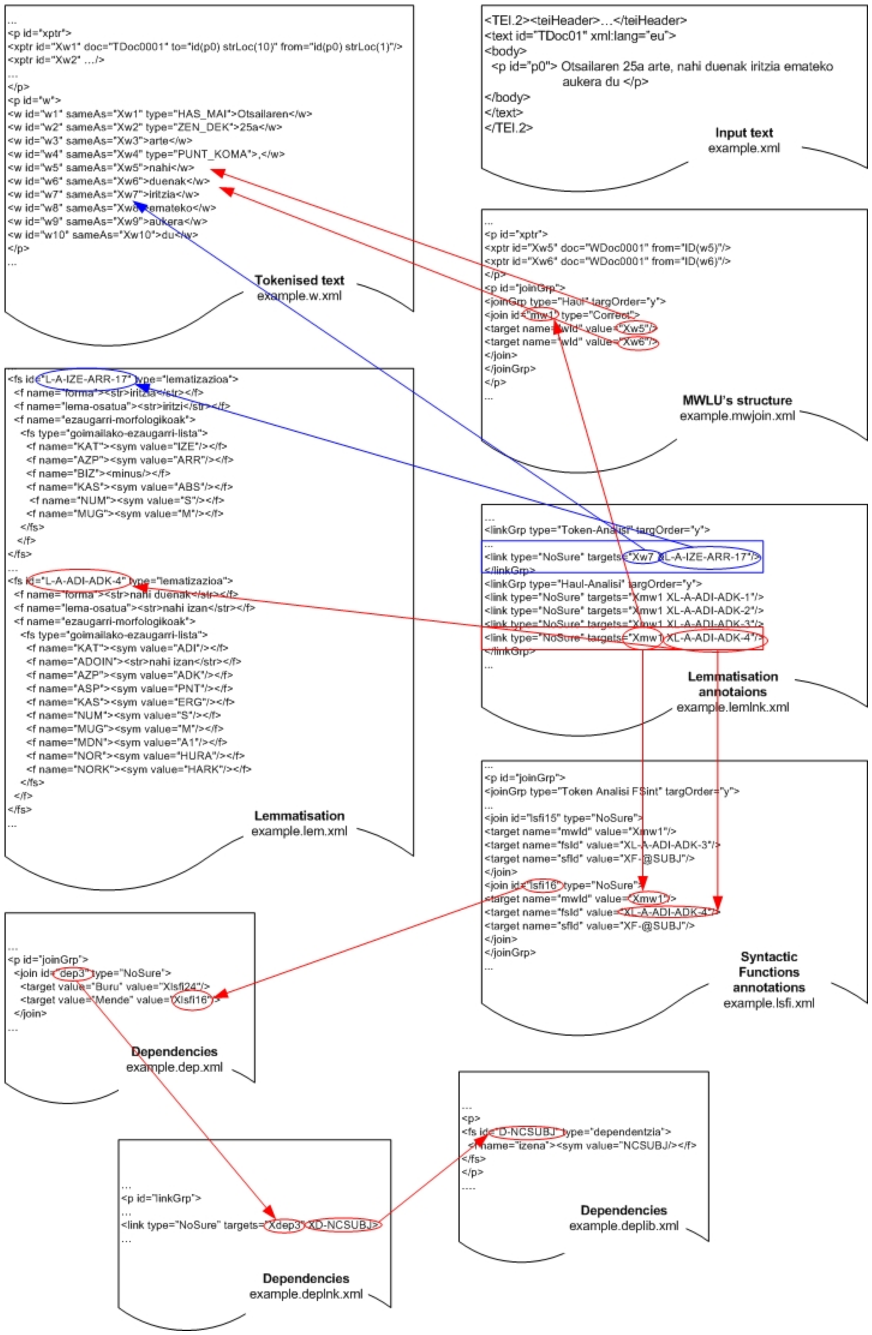
Within *Ixati*, the named entity recogniser and classifier named *Eihera* (Alegria *et al.*, 2003b) has a precision level of 78.99% and a rate of recall of 83.73% in the identification task. In the classification task, its precision level is 81.36%. *Ixati* has a precision level of 93.13% and a rate of recall of 88.73% in the task of tagging post-position expressions (Aranzabe, 2008). Finally, it has a precision level of 80.3% and a rate of recall of 94.7% when tagging noun chains of one component. For those with more than one component, the precision level is 84% and the rate of recall is 91.1%.

Finally, the in-depth analysis phase establishes the dependency-based grammatical relations between the components within the clause. The aim of the in-depth analysis is to establish the dependency relations between the components in the sentence. This process is performed by means of CG rules. The dependencies constitute a hierarchy that describes the most important grammatical structures such as relative clauses, causative sentences, coordination, discontinuous elements and so on.

The library LibiXaML (Artola *et al.*, 2009) is used to access all of the linguistic information obtained from the analysis. There is a data model which interprets the structure and relations of the information and it is represented by the classes encapsulated in LibiXaML. The annotation model relies on XML technologies for data representation, storage and retrieval.

Figure IV.2 shows an example of the output of the analyser after tokenisation, lemmatisation and dependency-based syntactic parsing. The input is the sentence Otsailaren 25a arte, nahi duenak iritzia emateko aukera du. (Those who want to do so have the opportunity to express their views until February 25th.). The example.w.xml stores the output of the tokenisation process. A multiword expression (nahi duenak (those who want to do so)) is represented within the structure document of the multiword lexical units (MWLUs) (example.mwjoin.xml). Annotations and ambiguities (if any) are represented by the link document (example.lemlnk.xml) that attaches the different items in the source text to their corresponding lemmas that are stored in example.lem.xml. Finally, as a result of the dependency-based syntactic parsing step, three more documents are created in the annotation web: example.dep.xml; example.deplib.xml; and example.deplnk.xml.[1] The syntactic functions, the token information and the lemmatisation information are related to the example.lsfi.xml document.

[1] We have not made use of the information relating to syntactic dependencies.



```
<TEI.2><teiHeader>…</teiHeader>
<text id="TDoc01" xml:lang="eu">
<body>
  <p id="p0"> Otsailaren 25a arte, nahi duenak iritzia emateko
                aukera du </p>
</body>
</text>
</TEI.2>
```

```
...
<p id="xptr">
<xptr id="Xw1" doc="TDoc0001" to="id(p0) strLoc(10)" from="id(p0) strLoc(1)"/>
<xptr id="Xw2" …/>
...
</p>
<p id="w">
<w id="w1" sameAs="Xw1" type="HAS_MAI">Otsailaren</w>
<w id="w2" sameAs="Xw2" type="ZEN_DEK">25a</w>
<w id="w3" sameAs="Xw3">arte</w>
<w id="w4" sameAs="Xw4" type="PUNT_KOMA">,</w>
<w id="w5" sameAs="Xw5">nahi</w>
<w id="w6" sameAs="Xw6">duenak</w>
<w id="w7" sameAs="Xw7">iritzia</w>
<w id="w8" sameAs="Xw8">emateko</w>
<w id="w9" sameAs="Xw9">aukera</w>
<w id="w10" sameAs="Xw10">du</w>
</p>
...
```

```
...
<p id="xptr">
<xptr id="Xw5" doc="WDoc0001" from="ID(w5)"/>
<xptr id="Xw6" doc="WDoc0001" from="ID(w6)"/>
</p>
<p id="joinGrp">
<joinGrp type="Haul" targOrder="y">
<join id="mw1" type="Correct">
<target name="wId" value="Xw5"/>
<target name="wId" value="Xw6"/>
</join>
</joinGrp>
</p>
...
```

```
<fs id="L-A-IZE-ARR-17" type="lematizazioa">
  <f name="forma"><str>iritzia</str></f>
  <f name="lema-osatua"><str>iritzi</str></f>
  <f name="ezaugarri-morfologikoak">
    <fs type="goimailako-ezaugarri-lista">
      <f name="KAT"><sym value="IZE"/></f>
      <f name="AZP"><sym value="ARR"/></f>
      <f name="BIZ"><minus/></f>
      <f name="KAS"><sym value="ABS"/></f>
       <f name="NUM"><sym value="S"/></f>
      <f name="MUG"><sym value="M"/></f>
    </fs>
   </f>
</fs>
...
<fs id="L-A-ADI-ADK-4" type="lematizazioa">
  <f name="forma"><str>nahi duenak</str></f>
  <f name="lema-osatua"><str>nahi izan</str></f>
  <f name="ezaugarri-morfologikoak">
    <fs type="goimailako-ezaugarri-lista">
      <f name="KAT"><sym value="ADI"/></f>
      <f name="ADOIN"><str>nahi izan</str></f>
      <f name="AZP"><sym value="ADK"/></f>
      <f name="ASP"><sym value="PNT"/></f>
      <f name="KAS"><sym value="ERG"/></f>
      <f name="NUM"><sym value="S"/></f>
      <f name="MUG"><sym value="M"/></f>
      <f name="MDN"><sym value="A1"/></f>
      <f name="NOR"><sym value="HURA"/></f>
      <f name="NORK"><sym value="HARK"/></f>
    </fs>
  </f>
</fs>
...
```

```
....
<linkGrp type="Token-Analisi" targOrder="y">
...
<link type="NoSure" targets="Xw7 XL-A-IZE-ARR-17"/>
</linkGrp>
<linkGrp type="Haul-Analisi" targOrder="y">
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-1"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-2"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-3"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-4"/>
</linkGrp>
...
```

```
...
<p id="joinGrp">
<joinGrp type="Token Analisi FSint" targOrder="y">
...
<join id="lsfi15" type="NoSure">
<target name="mwId" value="Xmw1"/>
<target name="fsId" value="XL-A-ADI-ADK-3"/>
<target name="sfId" value="XF-@SUBJ"/>
</join>
<join id="lsfi16" type="NoSure">
<target name="mwId" value="Xmw1"/>
<target name="fsId" value="XL-A-ADI-ADK-4"/>
<target name="sfId" value="XF-@SUBJ"/>
</join>
</joinGrp>
...
```

```
...
<p id="joinGrp">
  <join id="dep3" type="NoSure">
    <target value="Buru" value="Xlsfl24"/>
    <target value="Mende" value="Xlsfi16"/>
  </join>
...
```

```
...
<p id="linkGrp">
...
<link type="NoSure" targets="Xdep3 XD-NCSUBJ>
...
```

```
...
<p>
<fs id="D-NCSUBJ" type="dependentzia">
  <f name="izena"><sym value="NCSUBJ/></f>
</fs>
</p>
....
```

Figure IV.2: Output of the Basque analyser

The linguistic information offered by the analyser was used in all of the experiments based on Basque corpora. When the generation of items is based on morphosyntactic information, the output of *Ixati* is essential (cf., section V.3). When the system employs the semantic information contained within the words in the distractor generation task, the output of *Eustagger* is enough, but as the chunks are also represented in the question model, *Ixati* is also necessary (cf., section VI.4).

Example IV.1.1 shows the information given at chunk level. From the sentence Nire anaiarekin joan nintzen etxera (I went home with my brother), the analyser obtains three chains: two noun chains [Nire anaiarekin], [etxera] and one verb chain [joan nintzen].[2] In the given example, the chunks are distinguished by means of the tags %SIH, %SIB, %SINT, %ADIKATHAS and %ADIKATBU. Thus, while the tags %SIH and %SIB are used to express the beginning and end of a noun chain, the tag %SINT is added to the analysis of a word when it comprises a one-element noun chain. Finally, the same procedure is applied in the case of verb chains. The example focuses on the noun chain containing two elements.

Example IV.1.1

```
"<Nire>"<HAS_MAI>"
     "ni" IOR PERARR NI GEN NUMS MUGM ZERO HAS_MAI @IZLG> %SIH
"<anaiarekin>"
     "anaia" IZE ARR SOZ NUMS MUGM AORG @ADLG %SIB
"<joan>"
     "joan" ADI SIN PART BURU NOTDEK @-JADNAG %ADIKATHAS
"<nintzen>"
     "izan" ADL B1 NOR NR_NI @+JADLAG %ADIKATBU
"<etxera>"
      "etxe" IZE ARR BIZ- ALA NUMS MUGM @ADLG %SINT
```

When working at chunk level, the morphosyntactic information that corresponds to the chain is offered with the last element of the chain. For instance, the word anaiarekin (with brother) contains the morphological information of the noun chain [nire anaiarekin] (with my brother). It is a noun chain with the sociative mark.

[2] [with my brother] [home] [I went]

IV.1.1.2 Basque generator

Basque is an agglutinative language (see Appendix A) with a rich morphology. This is why there was a need to integrate a morphological generator within the distractor generation task when generating Basque items.

The morphological analyser developed by the IXA research group is also a morphological generator. Originally, the morphological analyser/generator was based on the application of a two-level morphology (Koskenniemi, 1983) for Basque, a well-known formalism for highly inflected and agglutinative languages.

The two-level system is based on a lexicon in which the morphemes and the possible links between them are defined using a set of rules which controls the mapping between the lexical level and the surface level due to the morphonological transformations involved. These rules have been compiled into transducers, meaning that it is possible to apply the system to both analysis and generation.

Nowadays, this Basque morphological grammar (the two-level morphology) has been ported (Alegria *et al.*, 2010) from Xerox formalism to the open-source *foma* toolkit (Hulden, 2009). Our system uses open-source grammar to introduce publicly available tools, as well as because of the recent shift in preference toward sequential replacement rules rather than two-level rules (Alegria *et al.*, 2010).

This morphological generator was used for all of the Basque items generated within the distractor generation task (see sections V.2, V.3 and VI.4).

IV.1.1.3 Connexor Machinese Syntax

Connexor Machinese software[3] products are NLP tools which analyse texts and provide information on the language content. This information is described using the Machinese Language Model. Connexor technology is based on linguistic methods and Functional Dependency Grammar (FDG) technology. Within its products, Machinese Syntax is a full-scale dependency parser. Machinese Syntax describes how different words and phrases relate to each other, that is, it carries out a syntactic analysis of sentences using dependency links that show the head-modifier relations between words. In addition, these links have labels that refer to the syntactic function of the modifying word. Therefore, Machinese Syntax offers detailed linguistic

[3] http://www.connexor.com

analysis which covers: tokenisation; decompounding; lemmas; proper nouns; part-of-speech (PoS); morphology; syntax and NPs.

Within the output of Machinese Syntax, the tree structure can be read in terms of either syntactic tags or syntactic relations. Syntactic tags express the structure of an NP in terms of premodifiers, postmodifiers and nominal heads, which together make an NP. Similarly, they label verbs as auxiliary verbs or the heads of verb phrases, which together make a complex verb. Syntactic relations go further and build complete trees out of NPs, verb phrases, adverbials and clause markers, in terms of subjects, objects, etc.

The creators of Connexor Machinese software have provided two evaluation results. In the analysis of standard written English, taken from the Maastricht Treaty, the accuracy of Machinese Syntax in terms of PoS is 99.3% with no ambiguity, and in the task of linking subjects and objects correctly (syntax accuracy), the level of precision is 93.5% and the rate of recall is 90.3%.[4] They also provide results for foreign news texts. As regards the syntactic accuracy of Machinese Syntax, its level of precision is 96.5% and its rate of recall is 95.4%. The authors claim that figures for different text types range between these two figures. Example IV.1.2 shows a sample of this output.

Example IV.1.2 (Machinese Syntax XML output)

```
<analysis>
  <sentence id="w:3">
    <token id="w:4">
      <text>This</text>
      <lemma>this</lemma>
      <depend head="w:5">subj:</depend>
      <tags>
        <syntax>@SUBJ %NH</syntax>
        <morpho>PRON DEM SG</morpho>
      </tags>
    </token>
    <token id="w:5">
      <text>is</text>
      <lemma>be</lemma>
      <depend head="w:3">main:</depend>
      <tags>
        <syntax>@+FMAINV %VA</syntax>
        <morpho>V PRES SG3</morpho>
      </tags>
    </token>
  </sentence>
```

[4] The Maastricht Treaty is legal text, which is a complicated genre to analyse.

```
 <paragraph/>
</analysis>
```

Every new sentence starts with the tag *sentence*, which has an identification value. The beginning of a token is expressed by the tag *token*. The *token* tag contains all of the linguistic information relating to the token. For instance, based on example IV.1.2, the token with the id "w:5" refers to the verb form "is". Together with the verb form, the model offers information on the lemma (*lemma*), morphology (*morpho*), syntax (*syntax*) and dependency (*depend*). Based on this information, we know that the verb form "is" is the main element function of the sentence and refers to the present singular third person.

In our experiments (cf., section VI.3), syntactic tags were used to build different components of the items. In other words, this information is necessary in order to build the correct stems and to create distractors with the same verb forms as the keys.

IV.1.2 Corpora

A corpus stores a collection of texts which are selected with one particular purpose. For this reason, some criteria and aims are first established, and the corpus is then built based on them to be a significant sample which matches the defined criteria.

There are different types of corpus (written or spoken, general or specific, monolingual or multilingual, synchronic or diachronic, raw or tagged) and they are used in different fields (NLP, lexicography, language teaching and learning and so on). In the following, the various corpora used by the system are presented.

IV.1.2.1 Euskaldunon Egunkaria newspaper corpus

The *Euskaldunon Egunkaria* newspaper corpus is a corpus that has been built up thanks to the collaboration between the IXA research group and the *Euskaldunon Egunkaria* and *Berria*[5] newspapers. The sample corpus in question comprises news from 2000, 2001, 2002 and 2004, with eight million words per year (on average).

[5] http://berria.info

The *Euskaldunon Egunkaria* newspaper corpus is based on news written solely in Basque, and so it is a monolingual, synchronic corpus. It is a structured corpus in which news stories are classified according to their date of publication, and they are analysed at the morphological level. The corpus is still growing, and in our experiments, we made use of the years analysed at the morphological level. The representation of the texts from 2004 is the same as is presented in Figure IV.2. However, 2000, 2001 and 2002 were analysed, but the representation of the information differs from the representation explained in the previous section. These years were analysed within the context of the HERMES project,[6] meaning that the XML representation used is the one defined in this project, as example IV.1.3 displays. Example IV.1.3 is an example that belongs to the year 2000.

Example IV.1.3 (*Euskaldunon Egunkaria* newspaper corpus)

```
<MW NETYPE="STRONG" FRM="Gobernu_Batzordeak">
  <CAT SCHEME="HERMES-MUC" CODE="RGANIZATION" />
  <LEX LEM="Gobernu_Batzordea"></LEX>
  <W FRM="Gobernu">
    <LEX LEM="gobernu" PAR="IZEARR"></LEX>
  </W>
  <W FRM="Batzordeak">
    <LEX LEM="Batzordea" PAR="IZEIZB"></LEX>
  </W>
</MW>
<W FRM="proposatutako">
  <LEX LEM="proposatu" PAR="ADI"></LEX>
</W>
```

Each new element is represented by means of the tags *MW* or *W*. The former is used to represent a multiword expression, while the latter is used for one-word elements. In both cases, the word form entry is accompanied by its lemma and PoS. For instance, the word form proposatutako (the suggested) (example IV.1.3) is a verb (*PAR="ADI"*) with the root or lemma (tag *LEMMA*) proposatu (to suggest).

As the *Euskaldunon Egunkaria* newspaper corpus was only used within the distractor generation task (see section V.3.1.1), the lemma and PoS of each word in the texts were enough to perform the task. This information was obtained by means of a simple XML parser included in ArikIturri.

[6] HERMES project: News databases. Cross-lingual information retrieval and semantic extraction (TIC-2000-0335-C03-03), founded by the Spanish Government. http://nlp.uned.es/hermes/

IV.1.2.2 Basque language learning corpus

The Ikasbil website,[7] which was developed by HABE, groups together various learning-oriented material: written texts; audio and video material; exercises etc., some of which are linked to one another.

Within this dissertation, we have built a Basque language learning corpus based on the written texts of Ikasbil. Therefore, this corpus is monolingual, specific and synchronic. It is specific because it contains texts that are focused on the learning process of Basque learners. Moreover, the corpus is classified into different language levels in accordance with the CEFR (cf., section II.3). Although the language level of a text can be a controversial aspect because it is difficult to define, in the corpus in question, expert teachers classified the texts into specific levels.

Table IV.1 shows the number of stored texts and sentences according to each language level. In total, the first version of the corpus stores 1303 texts (695,415 words). These are the numbers used in the study presented in section IV.1.4. Thus far, the corpus has remained open and Ikasbil is constantly enriching it with new texts.

Level	**#Texts**	**#Sentences**	**#Words**
A1-B1	713	23,094	361,158
B2	356	10,836	167,129
C1	234	10,079	167,128

Table IV.1: Basque language learning corpus

The texts have been analysed by means of *Ixati*, meaning that the obtained output type is the same as is shown in Figure IV.2. This corpus was used as the input for generating declension and verb tests (cf., section V.3.1).

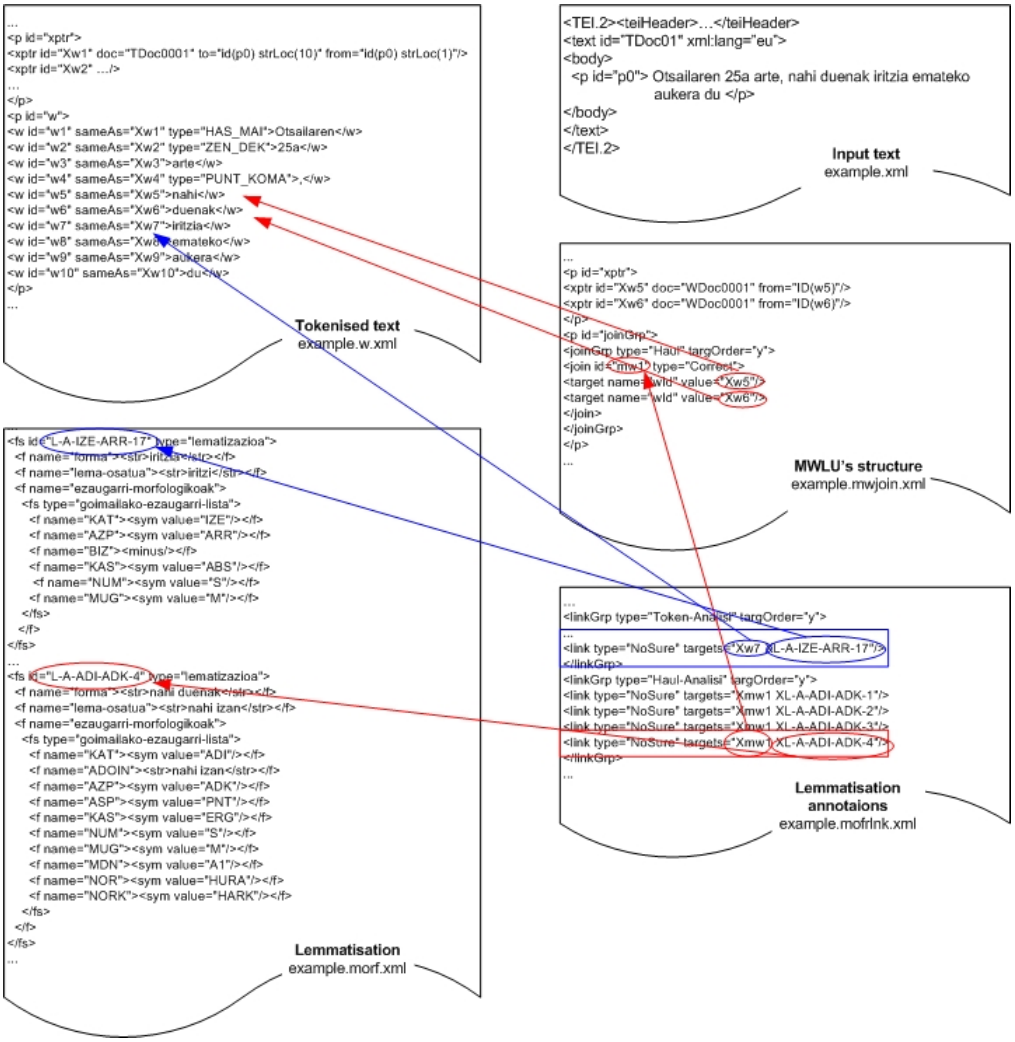
IV.1.2.3 ZT corpus

The *Zientzia eta Teknologiaren Corpusa*[8] (ZT corpus) (Areta *et al.*, 2007) is a collection of 8.5 million words which comprise a collection of Basque texts relating to science and technology. The ZT corpus is structured and tagged and is intended to be a reference resource for research on the use

[7] http://www.ikasbil.net/jetspeed/

[8] http://www.ztcorpusa.net: Science and Technology Corpus.

of Basque. This corpus is monolingual and synchronic because it contains works published between 1990 and 2002. It is also a specific corpus, due to the domain of the source texts, and it is classified according to the field of knowledge and genre of the text.



```
...
<p id="xptr">
<xptr id="Xw1" doc="TDoc0001" to="id(p0) strLoc(10)" from="id(p0) strLoc(1)"/>
<xptr id="Xw2" .../>
...
</p>
<p id="w">
<w id="w1" sameAs="Xw1" type="HAS_MAI">Otsailaren</w>
<w id="w2" sameAs="Xw2" type="ZEN_DEK">25a</w>
<w id="w3" sameAs="Xw3">arte</w>
<w id="w4" sameAs="Xw4" type="PUNT_KOMA">,</w>
<w id="w5" sameAs="Xw5">nahi</w>
<w id="w6" sameAs="Xw6">duenak</w>
<w id="w7" sameAs="Xw7">iritzia</w>
<w id="w8" sameAs="Xw8">emateko</w>
<w id="w9" sameAs="Xw9">aukera</w>
<w id="w10" sameAs="Xw10">du</w>
</p>
...
```

```
...
<fs id="L-A-IZE-ARR-17" type="lematizazioa">
  <f name="forma"><str>iritzia</str></f>
  <f name="lema-osatua"><str>iritzi</str></f>
  <f name="ezaugarri-morfologikoak">
    <fs type="goimailako-ezaugarri-lista">
      <f name="KAT"><sym value="IZE"/></f>
      <f name="AZP"><sym value="ARR"/></f>
      <f name="BIZ"><minus/></f>
      <f name="KAS"><sym value="ABS"/></f>
       <f name="NUM"><sym value="S"/></f>
      <f name="MUG"><sym value="M"/></f>
    </fs>
   </f>
</fs>
...
<fs id="L-A-ADI-ADK-4" type="lematizazioa">
  <f name="forma"><str>nahi duenak</str></f>
  <f name="lema-osatua"><str>nahi izan</str></f>
  <f name="ezaugarri-morfologikoak">
    <fs type="goimailako-ezaugarri-lista">
      <f name="KAT"><sym value="ADI"/></f>
      <f name="ADOIN"><str>nahi izan</str></f>
      <f name="AZP"><sym value="ADK"/></f>
      <f name="ASP"><sym value="PNT"/></f>
      <f name="KAS"><sym value="ERG"/></f>
      <f name="NUM"><sym value="S"/></f>
      <f name="MUG"><sym value="M"/></f>
      <f name="MDN"><sym value="A1"/></f>
      <f name="NOR"><sym value="HURA"/></f>
      <f name="NORK"><sym value="HARK"/></f>
    </fs>
  </f>
</fs>
...
```

```
<TEI.2><teiHeader>...</teiHeader>
<text id="TDoc01" xml:lang="eu">
<body>
  <p id="p0"> Otsailaren 25a arte, nahi duenak iritzia emateko
              aukera du </p>
</body>
</text>
</TEI.2>
```

```
...
<p id="xptr">
<xptr id="Xw5" doc="WDoc0001" from="ID(w5)"/>
<xptr id="Xw6" doc="WDoc0001" from="ID(w6)"/>
</p>
<p id="joinGrp">
<joinGrp type="Haul" targOrder="y">
<join id="mw1" type="Correct">
<target name="wId" value="Xw5"/>
<target name="wId" value="Xw6"/>
</join>
</joinGrp>
</p>
...
```

```
...
<linkGrp type="Token-Analisi" targOrder="y">
...
<link type="NoSure" targets="Xw7 XL-A-IZE-ARR-17"/>
</linkGrp>
<linkGrp type="Haul-Analisi" targOrder="y">
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-1"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-2"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-3"/>
<link type="NoSure" targets="Xmw1 XL-A-ADI-ADK-4"/>
</linkGrp>
...
```

Figure IV.3: ZT corpus example.

The ZT corpus is represented in XML and it has been linguistically tagged by means of *Eustagger*. The automatic tagging process provides the lemma, category and sub-category of each word. Finally, the corpus is divided into a balanced part and an open part. The balanced part comprises a sample of significant texts from the field of science and technology and, its PoS annotation was manually revised. The open part contains works collected according to their accessibility and, was automatically tagged.

Figure IV.3 shows an example of the part of the annotation net that is obtained at the morphological level. This corpus was used within the distractor generation task, as is explained in section VI.4.

IV.1.2.4 British National Corpus

The British National Corpus (BNC) (BNC Consortium, 2007) is a collection of 100 million words which make up samples of written and spoken language from a wide range of sources in order to represent a wide cross-section of British English from the latter part of the 20^{th} century. Therefore, the BNC is a general corpus because of its variety of fields, genre and registers. It is also a monolingual corpus which deals with modern British English; however, as it only covers British English from the late 20^{th} century, it is a synchronic corpus. Most (about 90%) of the words are taken from many kinds of written texts and 10% are taken from transcribed speech.

The BNC was preprocessed by our system in order to obtain a tagged corpus represented in XML. This was obtained by parsing the BNC using Connexor Machinese Syntax. Example IV.1.4 shows part of text J10[9] (a novel by Michael Pearce)[10] that corresponds to the sentence But, ' said Owen, 'where is the body?'. The word "said" in this example is the main function of the sentence, while "Owen" is the subject of the sentence. Therefore, the *depend* tag' **head** attribute value is "w10", the id that corresponds to the word "said".

Example IV.1.4 (BNC example)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE analysis SYSTEM "http://www.connexor.com/dtds/4.0/fdg3.dtd">
```

[9] Data cited herein have been extracted from the British National Corpus, distributed by Oxford University Computing Services on behalf of the BNC Consortium. All rights in the texts cited are reserved.

[10] This is one of the examples given in the Reference Guide for the British National Corpus (XML Edition). This is an easy way to compare the output of the BNC XML edition and Machinese Syntax output.

```
<analysis>
 ...
  <token id="w10">
    <text>said</text>
    <lemma>say</lemma>
    <depend head="w5">main:</depend>
    <tags>
     <syntax>@+FMAINV %VA</syntax>
     <morpho>V PAST</morpho>
    </tags>
  </token>
  <token id="w11">
    <text>Owen</text>
    <lemma>owen</lemma>
    <depend head="w10">subj:</depend>
    <tags>
     <syntax>@SUBJ %NH</syntax>
     <morpho>N NOM SG</morpho>
    </tags>
  </token>
....
</analysis>
```

In the given sentence, there is a word ("where") for which the analyser was unable to decide its corresponding syntactic analysis as example IV.1.5 shows.

Example IV.1.5 (BNC example — Ambiguous case)

```
  <token id="w14">
    <text>where</text>
    <lemma>where</lemma>
    <tags>
     <syntax>@ADVL %EH</syntax>
     <morpho>ADV WH</morpho>
    </tags>
    <tags>
     <syntax>@<P %EH</syntax>
     <morpho>ADV WH</morpho>
    </tags>
  </token>
```

In our approach, this type of ambiguous case is not addressed, and the first possible analysis is considered to be the correct one.

The corpus was used in different steps of the generation process, as section VI.3 will show.

IV.1.2.5 Web 1T 5-gram dataset

The Web 1T 5-gram dataset (Brants and Franz, 2006) is not a traditional corpus, because it contains English word n-grams and their frequency counts. However, this is also a useful resource for statistical language modelling, among other uses. The length of the n-grams ranges from one to five.

The n-gram counts were generated from approximately one trillion word tokens of text taken from publicly-accessible Web pages collected in January 2006. The authors intended to use only Web pages with English text, but the data also contain some text in other languages. The preprocess used to obtain the corpus involved character encoding, tokenisation and filtering. In addition, all tokens which appear at least 200 times are offered, as well as n-grams appearing 40 times or more.

The following is an example of the 3-gram data contained in this corpus:[11]

ceramics collectables collectibles	55
ceramics collectables fine	130
ceramics collected by	52
ceramics collectible pottery	50
ceramics collectibles cooking	45

The following is an example of the 4-gram data contained in this corpus:

serve as the incoming	92
serve as the incubator	99
serve as the independent	794
serve as the index	223
serve as the indication	72
serve as the indicator	120
serve as the indicators	45
serve as the indispensable	111

This information was used for the sentence selection and distractor generation tasks in section VI.3.

[11] Details can be found at http://www.ldc.upenn.edu/Catalog/docs/LDC2006T13/

IV.1.2.6 Basque learner corpus

Learner corpora are collections of texts produced by foreign/second language learners. Maritxalar (1999) started collecting texts, and through this, the IXA research group has built up a Basque learner corpus (Uria, 2009). This is a specific corpus, as it contains texts from one type of text producer. It is monolingual because it is a collection of Basque essays.

This type of corpus is essential for detecting and predicting the difficulties that learners will experience, as well as developing specific tools to help them in the learning process. The collected essays are the source which is used to identify the errors made by students and the difficulties they have. These texts offer a way to detect the variations between students from one language level to another, and so on.

In addition, this type of corpus is also interesting with regard to creating exercises based on learners' errors. This type of information can be very useful for creating items based on learners' performance. In this way, the corpus can be used to select the source sentences as well as to generate distractors based on learners' errors. Having a system which creates items based on learners' performance give teachers the option of not correcting students' errors directly, but proposing tests based on their errors. This is a way of studying whether one particular student has a problem with one particular phenomenon or detecting some casual errors which can be corrected once the learner sees them.

Level	**#Texts**	**#Words**
A1-B1	300	39,117
B2	207	42,219
C1	129	31,954
Total	636	113,290

Table IV.2: Basque learner corpus

Our Basque learner corpus is composed of texts from different Basque learning schools, academic years, language levels, students and types of essays. At this time, the corpus consists of 113,290 words which are divided into three language levels: low, middle and high. These levels corresponds to the A1, A2, B1, B2 and C1 levels of the CEFR. The reason for making the former distinction was that before the CEFR was established, the levels were distinguished this way in HABE (1999). Table IV.2 shows the number of words for each language level (Uria *et al.*, 2009), in accordance with the CEFR.

Each text is identified with a unique code which comprises the source of the text, the year, the language level, the learner's identification and the exercise type. This information is stored in order to help during the linguistic and psycholinguistic diagnosis process (Uria, 2009).

We analysed the entire corpus at the chunk level by means of *Ixati*. Example IV.1.6 shows a sample of a possible learner text in which there is a grammatical error at the chunk level. More specifically, there is a determiner error, and the obtained analysis is incorrect in the given context.

Example IV.1.6 (Learner corpus example)

```
"<Txakurra>"<HAS_MAI>"
      "txakur" IZE ARR ABS NUMS MUGM HAS_MAI @OBJ|@PRED %SINT
"<bat>"
      "bat" DET DZH NMGS ABS MG @OBJ|@PRED %SINT
"<ikusi>"
      "ikusi" ADI SIN PART BURU NOTDEK @-JADNAG %ADIKATHAS
"<nuen>"
      "*edun" ADL B1 NOR_NORK NR_HURA NK_NIK @+JADLAG %ADIKATBU
"<.>"<PUNT_PUNT>"
     PUNT_PUNT
```

Example IV.1.7 shows the analysis of the correct sentence.

Example IV.1.7 (Learner corpus example — Corrected)

```
"<Txakur>"<HAS_MAI>"
      "txakur" IZE ARR ZERO HAS_MAI @KM> %SIH
"<bat>"
      "bat" DET DZH NMGS ABS MG @OBJ|@PRED %SIB
"<ikusi>"
      "ikusi" ADI SIN PART BURU NOTDEK @-JADNAG %ADIKATHAS
"<nuen>"
      "*edun" ADL B1 NOR_NORK NR_HURA NK_NIK @+JADLAG %ADIKATBU
```

There is a difference in terms of chunk detection between the two examples. Example IV.1.6 contains three chains [Txakurra] [bat] [ikusi nuen],[12] while example IV.1.7 has two [Txakur bat] [ikusi nuen].[13] This is due to

[12] [A dog][one][I saw]

[13] [A dog][I say]

the extra determiner in example IV.1.6. We have to take into account that the morphosyntactic analyser has been developed in order to analyse correct texts, and so when there is a determiner error, the analyser does not mark it and it is unable to detect the correct chain. Different works by IXA have focused on the detection and correction of errors (Uria, 2009; Oronoz, 2009). Uria (2009) defined a CG system for detecting determiner errors, while Oronoz (2009) presented different tools for working with date-errors, postposition expression errors and agreement errors. Starting from the output of *Ixati*, these tools can be used by our system to generate items.

In our experiments (cf. section V.3.2), we worked with determiner errors, so the system employs the CG proposed by (Uria, 2009). The rules created for the automatic detection of determiner errors provided the basis for generating distractors and exercises relating to the correct and incorrect use of determiners.

IV.1.3 Ontologies and dictionaries

ArikIturri not only employs morphosyntactic information to generate test items, but also semantic information. In particular, semantic information is used within the distractor generation task (cf., chapter VI). For this reason, the system takes advantage of different resources that offer semantic information relating to words. These resources are dictionaries and ontologies.

IV.1.3.1 Dictionaries

Dictionaries, in general, offer the chance to find information relating to specific words. An entry in a dictionary usually comprises the word, its definition, the PoS, the pronunciation, related forms of the word and its origin. In addition, some specialised dictionaries exist in which additional information is provided, for instance, information relating to the topic.

In our experiments, the two dictionaries presented here were used in the generation of domain-specific test items in Basque (cf., section VI.4) and so the dictionaries presented here are Basque dictionaries.

Monolingual dictionary

The system makes use of the work by Díaz de Ilarraza *et al.* (2002), in which semantic features of common nouns are extracted semi-automatically from a monolingual dictionary. For this purpose, the researchers first used the information about genus data,[14] specific relators and synonyms extracted by Agirre *et al.* (2000) from the definitions contained in the monolingual dictionary entitled *Euskal Hiztegia* (Sarasola, 1996).

In order to label the common nouns that appear in the dictionary, the authors used the definitions of the 26,461 senses of the 16,380 common nouns defined by means of genus/relators (14,569) or synonyms (11,892). First, they labelled the semantic features of a small number of words in order to infer the value of the features for other words. Then, they expanded the labelling using synonyms, as well as heritage through the genus' hypernymy relationship.

The authors show the evaluation results regarding the ±[animate] feature, with an overall accuracy rating of 99.2%. The scope for the automatic labelling of the feature is 75.14% of all the nouns contained in the dictionary (12,308 of 16,380). In addition, they also present the results in a real context, using 311,901 common nouns, of which 7,219 are different, from the *Euskaldunon Egunkaria* newspaper corpus, with a scope of 69.2%.

In Díaz de Ilarraza *et al.* (2002), the features which are addressed are: ±[animate]; ±[human]; and ±[concrete]. Thus far, this information has been extended, and our system takes into account the ±[animate], ±[language], ±[time], ±[material], ±[device,vehicle], ±[communication-tool] and ±[measure] features of more than 15,000 entries in the experiments presented in section VI.4 within the distractor generation task.

Encyclopedic dictionary of science

Zientzia eta Teknologiaren Hiztegi Entziklopedikoa[15] (Elhuyar Hizkuntza Zerbitzuak, 2009) is an encyclopedic dictionary of science and technology that was developed by the Elhuyar Foundation.[16] This non-profit foundation detected the need for such a resource in the Basque scientific community, and aimed to produce a reference dictionary in the field for the Basque language.

The interesting feature of this dictionary is the fact that the dictionary entries are distinguished by their domain. The dictionary comprises 23,000 basic concepts relating to science and technology, divided into 50 different

[14] The genus is usually the core of a definition sentence (Agirre *et al.*, 2000).

[15] Encyclopedic Dictionary of Science and Technology

[16] http://www.elhuyar.org

subjects organised according to six groups: **(i)** exact sciences (mathematics, statistics); **(ii)** matter and energy sciences (physics, chemistry, astronomy); **(iii)** Earth sciences (geography, geology, mineralogy, oceanography, paleontology); **(iv)** life and health sciences (biology, microbiology, botanics, zoology, biochemistry, genetics, physical anthropology, ecology, the environment, anatomy, physiology, medicine, psychiatry, veterinary science); **(v)** technology (technology, mechanical technology, electric technology, electronics, telecommunications, informatics, materials, architecture, construction, stock breeding, agriculture, fishing, mining, aeronautics, astronautics, sea, railways, automobile construction, photography, the arms industry); and **(vi)** general.

The dictionary also offers the entries in English, Spanish and French in order to connect Basque with these languages. Although this feature has not been exploited in this dissertation, it could be useful for making use of resources in those languages. The domain feature of each entry is exploited in section VI.4 in order to generate distractors.

IV.1.3.2 Ontologies

Within the context of information sciences, Gruber (2009) defines an ontology as a set of representational primitives which can be used to model a domain of knowledge or discourse. These representational primitives are classes, attributes and relationships, and their definitions include information about their meaning and constraints on their logically consistent application. The semantic interpretation of language requires an extensive and rich lexical knowledge base (LKB).

WordNet and Multilingual Central Repository

In the NLP field, one of the most well-known LKBs is *WordNet* (Fellbaum, 1998), a large lexical database of English developed at Princeton University. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each one corresponding to a single lexical concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. When more than one lexical unit share the same meaning, they are grouped together into a synset (synonymy). In addition, *WordNet* contains among others hypernymy and hyponymy relations. Example IV.1.8 shows the three different senses of the English noun science.

Example IV.1.8 (Senses of the term science)

***Sense 1**: science, scientific knowledge (any domain of knowledge accumulated by systematic study and organized by general principles)*

***Sense 2**: science, scientific discipline (a particular branch of scientific knowledge)*

***Sense 3**: science, skill (ability to produce solutions in some problem domain)*

If we look at the second sense of the noun (a particular branch of scientific knowledge), the hypernymy relation shows that one of the sense of the word discipline is a hypernym of science. Hyponymy is the inverse relation. For instance, science is a hyponym of natural science. Therefore, a synset can be seen as the semantic class that groups together a complete set of hyponyms. In the case of verbs, troponymy is used to encode the hierarchy of verbs. "Verb Y is a troponym of the verb X if the activity Y is doing X in some manner" (Pociello *et al.*, 2010).

EuroWordNet (Vossen, 1998) is a multilingual database that comprises WordNets for eight European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). It follows the same model as Princeton's WordNet, and the WordNets are linked to an Inter-Lingual-Index (ILI). Therefore, the WordNets are interconnected, but each one can be seen as an independent WordNet. Compared to *WordNet*, *EuroWordNet* includes new features: domain ontologies, a top ontology and base concepts.

The domain ontology groups synsets in terms of topics, while the Top Concept Ontology provides a common framework for the most important concepts in all of the WordNets which reflects important semantic distinctions. Finally, the base concepts are those which play the most important role in the various WordNets in different languages. The aim of this resource is to achieve the maximum degree of overlap and compatibility between WordNets in different languages and to allow the distributive development of WordNets across the world.

Finally, the MCR follows the *EuroWordNet* model and integrates various local wordnets (Basque, Catalan, English, Italian and Spanish) with different ontologies (Atserias *et al.*, 2004). The WordNets are enriched with a new kind of information: WordNet Domains; the Suggested Upper Merged Ontology

(SUMO) (Niles and Pease, 2001); and selectional preferences (Agirre and Martinez, 2002).

The Basque WordNet (Pociello *et al.*, 2010) contains 32,456 synsets and 26,565 lemmas, and it is complemented by a hand-tagged corpus comprising 59,968 annotations.

As previously noted, these resources are used in section VI.4 within the distractor generation task. As the information is in different languages, the problem of insufficient semantic information for the Basque language is resolved. In this approach, the system takes into account the properties of the Top Concept Ontology, the WordNet Domains and the SUMO.

IV.1.4 Analysis

We have already mentioned that the two main resources used by the system to generate exercises are corpora and NLP tools. In all of the generated tests presented in chapters V and VI, a corpus was used as the starting point from which to select the stems for the generation process of the distractors. Therefore, different corpora are used for different purposes in different steps of the system. In the case of NLP tools, a study of the influence of the tools in the generation process is also presented to see if the quality of the tools can have any influence on the results of the system.

IV.1.4.1 Corpora as a source of questions

In our experiments, the *sentence retriever* module (cf., section II.4.3) selected candidate sentences from the source corpus. This input corpus varied depending on the experiment.

The experiments which focused on generating items for a Basque language learning environment predominantly made use of two different corpora: a Basque language learning corpus and a Basque learners' corpus. Both corpora are useful in a language learning environment. The former contains texts selected by teachers according to the relevant language level. The latter comprises real texts written by students.

Regarding the tests for an English language learning environment, there was no specialised corpus nor a learner corpus available. As the BNC is used in various classrooms to find examples of use, the tests for the English language came from a sample of the BNC. In addition, a mystery/detective novel comprised of 140,000 words was also used to compare the general domain of the BNC with a specific domain — the domain of the novel. In addition to the initial selection of the sentences, we worked with the idea of offering as the stem of the item a sentence which comprises one of the most frequent collocations of the verb. For this reason, we extracted patterns of occurrences from the Web 1T 5-gram dataset.

Finally, when generating the tests for the science domain, the Basque texts were selected from a website[17] that provides current and up-to-date information on science and technology in Basque. In this case, the experiment was more focused on the study of the techniques for generating distractors automatically.

We have mentioned that the Basque language learning corpus is a classified corpus in which texts correspond to a specific language level. Thus, we could assume that, depending on the language level and the curriculum, certain linguistic phenomena should or should not appear. In order to test this theory, we carried out the following study.

Study of the influence of the Basque language learning corpus on the generation process

In order to corroborate the assumption that, depending on the language level, some linguistic phenomena should appear more frequently than others at each level, we analysed the Basque language learning corpus. More specifically, we studied two measures for each level: the frequency of each linguistic phenomenon defined in the HEOK (see Appendix C) and their respective importance/weight. The results presented in this dissertation were obtained during the development of the system. In other words, we have studied the influence of the selected corpus during the generation process of the declension and verb tests (cf., section V.3.1), as the input for those tests was the Basque language learning corpus. The topics of these tests are part of the linguistic phenomena defined in the curricula of Basque language schools.

The analysis of the corpus as a whole shows that some of the linguistic phenomena which are taught at the language schools at different levels do not appear in the corpus. Moreover, the percentage of appearances of the phenomena in the corpus does not match the level of importance that teachers have assigned to learning this linguistic content. Table IV.3 shows the five

[17] http://zientzia.net

declension cases which appear most often, as well as the percentages of the three verb tenses which occur at least once.[18]

Phenomenon	**Total**	**A1-B1 levels**	**B2 level**	**C1 level**
Sociative	3074 (6.98%)	1594 (6.90%)	777 (7.17%)	703 (6.97%)
Inessive	17,562 (39.90%)	9083 (39.33%)	4327 (39.93%)	4152 (41.19%)
Dative	4521 (10.27%)	2321 (10.05%)	1052 (9.71%)	1148 (11.39%)
Absolutive	33,383 (75.85%)	17,369 (75.21%)	8130 (75.03%)	7884 (78.22%)
Ergative	9115 (20.71%)	4704 (20.37%)	2190 (20.21%)	2221 (22.03%)
Present indicative-DA[19]	11,972 (27.20%)	6232 (26.98%)	2807 (25.90%)	2933 (29.10%)
Present indicative-DU	4490 (10.20%)	2351 (10.18%)	1032 (9.52%)	1107 (10.98%)
Present indicative-DIO	215 (0.49%)	116 (0.50%)	45 (0.41%)	54 (0.53%)
Present indicative-ZAIO	121 (0.27%)	62 (0.27%)	26 (0.24%)	33 (0.33%)
Past indicative-DA	242 (0.55%)	122 (0.53%)	57 (0.53%)	63 (0.62%)
Past indicative-DU	146 (0.33%)	77 (0.33%)	35 (0.32%)	34 (0.34%)
Past indicative-DIO	14 (0.03%)	7 (0.03%)	3 (0.03%)	4 (0.04%)
Past indicative-ZAIO	4 (0.01%)	2 (0.008%)	2 (0.02%)	0 (0%)
Present conditional-DA	127 (0.29%)	65 (0.28%)	30 (0.28%)	32 (0.32%)
Present conditional-DU	55 (0.12%)	27 (0.12%)	13 (0.12%)	15 (0.15%)
Present conditional-DIO	2 (0.004%)	1 (0.004%)	1 (0.009%)	0 (0%)
Present conditional-ZAIO	3 (0.006%)	2 (0.008%)	0 (0%)	1 (0.009%)
#Sentences	44,009	23,094	10,836	10,079

Table IV.3: Corpus analysis

There is no significant difference between the first two levels as regards the percentage of appearances of the different linguistic phenomena. In general, these differences are not statistically significant ($p.value > 0.05$). On the contrary, the differences between the higher level and the other two levels are almost always statistically significant ($p.value < 0.05$). In any case, the frequency of each linguistic phenomenon is similar across all language levels. This is unexpected, if we consider that the higher the language level, the higher the number of linguistic expressions a learner should acquire. This phenomenon could be caused by the fact that the texts offered in Ikasbil are more focused on the communication skills of the learners than on their grammatical ability.

What the corpus achieves is that all of the declension and verb tenses which appear at each level must be known by learners, as defined in HEOK (see Appendix C). For instance, at the B2 level, learners must know the

[18] We have distinguished these values for each verb paradigm.

[19] See Appendix A

Phenomenon	C1 level
Sociative	703
Inessive	4152
Dative	1148
Absolutive	7884
Ergative	2221
Present indicative	4127
Past indicative	101

Table IV.4: Corpus analysis

present indicative and it occurs a total of 3910 times. In contrast, there are no occurrences of the past conditional tense, which is not a requirement of this particular level.

However, the results of the analysed texts show that it is not possible to work with all of the cases of morphological inflection and verb conjugation. For example, although learners at the C1 level have to know the past conditional, it cannot be studied if the system selects candidates from the Basque language learning corpus, because there are no occurrences of the past conditional.

Therefore, based on the results, the importance of the distinction between different language levels for our experiments is not clear.

In the end, the system generated items using only the high language level corpus to avoid the noise that teachers would generate when discarding questions at lower levels because of the difficulty that students may experience with understanding the isolated sentences. In addition, linguistic phenomena with a percentage of occurrence lower than 0.2% were discarded. Thus far, we have chosen five inflection cases and four different verb forms corresponding to different paradigms, modes, aspects and tenses. Table IV.4 summarises these topics.

Finally, this type of corpus is useful for working with some types of grammar, but for other types, we would need to create sentences artificially or to apply our heuristics to a more general corpus, such as the *Euskaldunon Egunkaria* newspaper corpus.

It is also worth noting the similar study carried out by Uria (2009). This study was based on the occurrence rates of errors, and the results led us to base our experiments in section V.3.2 on a particular type of determiner error, that is, on the repetition of the determiner in the determiner phrase. The results of the analysis are presented in section V.3.2.

In conclusion, there are experiments in which the input corpus can influence the results. In such cases, an analysis of the corpus was carried out before the heuristics (which form the basis for dealing with the topics) were defined.

IV.1.4.2 Corpora within the distractor generation task

As we will explain in the following chapters, there are different ways to generate distractors and different selection criteria for candidates. One of the options is to make use of the information offered by corpora. In this section, we present corpora as a measure of similarity and as a source when searching for occurrences of use.

Corpora as a measure of similarity

Within the distractor generation task, various corpora are used to measure the similarities between words when semantic information is required. In other words, the corpus is used to build a language model which is then used to measure the distributional similarity of words.

Distributional similarity measures are based on the idea that the similarity between two words depends on the commonalities between their contexts. Thus, two words are similar if they occur in similar contexts. This type of similarity measures has been studied in different domains, meaning that different corpora have been used to compute the measures. As regards the tests generated for the science domain (cf., section VI.4), ArikIturri made use of the ZT corpus to measure similarities. In contrast, when working with English verbs (cf., section VI.3), the system made use of the BNC.

Based on the available tools, we decided the following:

- In domain-specific tests, the corpus is used to build a semantic model.
- For English verbs, we computed the information radius measure based on distributional data from the BNC.

In order to obtain the information radius measure (Dagan *et al.*, 1997) for an input word and based on predicate-object co-occurrence pairs, the tool retrieved the most similar words. This method has obtained good results in other application domains, and so we applied it to the English tests. However, it was not possible to apply this method to the Basque tests because of the lack of this type of co-occurrence pair. We carried out an experiment in order to obtain predicate-object co-occurrence pairs, but the information obtained was not enough to measure similarities in this way, as the EPEC[20] corpus, the only available corpus containing predicate-object pairs, contains 300,000 words. Therefore, for Basque, we built a model based only on information relating to the lemma and category of words. Therefore, the measure which was applied in each scenario was selected based on the availability and appropriateness of the corpus.

The specialised corpus, the ZT corpus, is a good option for model-building when working in the science and technology domain. We have already mentioned that the ZT corpus is composed of a balanced part and an open part. For this work, we used the balanced part (1.9 million words) of the specialised corpus because we consider the use of a balanced part to be more important than the use of a bigger corpus. We consider the semantic model built based on the balanced part to be more representative, because this part was developed based on different predefined criteria.

The system made use of the ZT corpus because of its availability. However, in Basque, as with any minority language, the construction of resources is difficult and expensive. Therefore, one might think that using this specialised corpus would be expensive due to the manual revision required, an expense that, depending on the situation, may not be feasible. If we look at the original tags for the balanced part of the corpus, there were originally around 260,000 ambiguous analyses. In total, 160,500 words were automatically disambiguated, and so manual work was carried out on 99,500 ambiguous words. This manual work was used to adapt the model of the tagger to the science domain and to develop better models to tag the rest of the corpus.

When generating distractors for a general domain, the system made use of domain-general corpora. There are some works in which the distractors are generated in the same way as they are in this dissertation (Mitkov *et al.*, 2009). These works have proven the appropriateness of using the BNC to measure similarities when making predicate-object co-occurrence pairs. In our approach, after similarity measures were applied, some innovations were

[20] Euskararen Prozesamendurako Erreferentzia Corpusa

proposed (cf., section VI.3).

In general, using different types of corpus to generate distractors shows the adaptability of the system to different scenarios. This is due to its general and flexible architecture (cf., section II.4). Therefore, this system could use new corpora, and could be incorporated as a source from which to generate distractors easily.

Corpora to search for occurrences

In addition to using corpora to measure similarities, we decided to exploit the available corpora in the distractor generation task in order to search for occurrences of use. In some experiments, this information is used to ensure that a candidate distractor is not a plausible answer in a given stem. In others, it is use to ensure that automatically generated inflected words exist.

As regards English verbs, the system searches for occurrences in the BNC as well as in the Web 1T 5-gram dataset. The former is used to extract patterns. The latter is used to obtain a language model which predicts the probability of the occurrence of a word sequence. The experiments regarding these matters are presented in section VI.3.

In the case of Basque language tests, the *Euskaldunon Egunkaria* newspaper corpus is used to search for examples of use. In this case, the corpus is used to search for occurrences of the candidate distractor's inflected form and as part of the criteria to generate the distractor. This process will be explained in chapter VI.

IV.1.4.3 Study of the influence of the NLP tools on the generation process

We have already mentioned that the selected corpus can have an influence on different steps of the generation process. The fact that NLP tools are also a main component of the system has already been explained. Therefore, the NLP tools which are used could also influence the results. In the following, we present the analysis in relation to the NLP tools at different steps of the generation process.

Topic identification and sentence selection

Topic identification is one of the primordial tasks of our system. This identification can be performed in different ways: based on a list of words; based on linguistic information; based on a term extractor and so on. Depending on the type of test, our system bases the identification task on different criteria.

When the scenario is an English language learning environment, the aim is to work with verbs from the AWL because we aim to generate tests to be used in a real scenario. Therefore, the starting point for identifying the topic is a list of verbs which do not comprise any difficulty in terms of topic identification. The system has to correctly identify the lemma and category of the verbs in the source sentence, something that is not difficult if we look at the 99.3% accuracy rate of Machinese Syntax regarding PoS tagging.

When dealing with Basque language tests, the results of the system depend very much on the match between the linguistic information regarding the answer focuses of the question and the specific topic that the teachers want to test, as the information used by the system is exclusively grammatical. Therefore, when working with NLP tools within this scenario, the robustness of these tools undoubtedly determines the results. The results depend, in some way, on the quality and sophistication of the parsers and generators.

As mentioned in section II.4.3, the *sentence retriever* module is responsible for detecting and selecting sentences in which the relevant topic to deal with appears. If any linguistic phenomena are not detected by *Morfeus*, then the system is unable to work with them. When we started developing the system, for instance, *Morfeus* did not correctly analyse the inflection of demonstrative pronouns. Nonetheless, *Morfeus* is periodically updated and improved so that the number of incorrectly detected phenomena is constantly decreasing.

When grammatical information is the basis for establishing the answer focuses of the items, the *answer focus identifier* module needs unambiguous information in order to be sure about the grammatical information. If a phrase is ambiguous, the identifier does not consider that phrase as a candidate answer focus. In contrast, if semantic information is required for the identification of the focuses, the ambiguous cases are also considered as candidates and the first analysis is established as the correct one.

When the system takes into account the semantic information regarding the words in the science domain, the aim is to start from a text and to work with its vocabulary. This is why the system needs to know which terms from the source text are meaningful. In order to detect these terms in an automatic way, the system could incorporate a term extractor for Basque based on Erauzterm (Alegria *et al.*, 2004). Erauzterm extracts the terminology of an entire corpus, while our system aims to extract terms from individual texts. Therefore, in addition to integrating Erauzterm into our system, it would also be necessary to modify it. Due to the lack of time, this improvement has not yet been developed and will be considered in future work (see chapter VII). We tried to employ the output of Erauzterm directly for each selected text, but the results of the term extractor were not satisfactory. As the aim of our experiments is to focus on the quality of the distractors, the extraction of meaningful terms from the source texts in this dissertation was carried out manually. For this reason, experts in the field took part (cf., section VI.4.1). However, a first attempt to base the generation of the items on the terms which were detected and classified automatically is presented as part of the transformation of declarative statements into interrogative ones in section V.2.

Distractor generation

As shown in the tasks presented above, the distractor generation process can be performed in multiple ways: by selecting the candidate distractors from a list; generating candidates based on the lemma of the correct answer but which are morphologically different; based on corpora; WordNets and so on.

In our approach, the fact that the candidate distractors are almost always presented in an automatically generated inflected form does not matter to the chosen criteria. Therefore, the correct operation of the integrated generation tools is indispensable.

This is not a problem in the case of English words, as all of the possible word formations are stored in a database. On the contrary, Basque is an agglutinative language with a rich morphology. For Basque, the system integrates the morphological generator which was presented above (cf., section IV.1.1.2) in order to create the corresponding inflected form.

Using grammatical information within the distractor generation task carries the risk that the generation will be based on verb conjugation and morphological declension tools. If the generation tools do not produce any output for the given input parameters, distractors will not be produced.

Another problematic point which was previously mentioned in section II.4.4 occurs when more than one of the generated distractors are identical, even if their morphological information is different. Example IV.1.9 shows a rejected MCQ. The *ill-formed question rejecter* module rejects the question because there are two identical distractors, i.e., (b) and (c), for different inflected forms, and the applied heuristics are unable to generate more distractors for this topic.[21]

Example IV.1.9 (Rejected MCQ example)

Drogazale hitza heroinaren menpe bizi diren .... uztartu izan da urte askoan.
The term drug addict has been related .... that live under the heroin's control.
*a) pertsona (lemma - people (*distractor*))*
*b) pertsonarekin (sociative definite singular - to the person (*distractor*))*
*c) pertsonarekin (sociative indefinite - to some people (*distractor*))*
*d) pertsonekin (sociative definite plural - to the people (*key*))*

In addition to using grammar in the generation of distractors, the system attempts to use semantic information in some experiments (cf. section VI). These experiments are focused on MCQs, as the automatic generation of distractors is the key point. Therefore, the NLP tools, techniques and resources which are used vary depending on the experiment. In addition, this is one of the main points of section VI, in which the results are fully expounded.

Section IV.1 has shown the resources available for the generation of items. Depending on the scenario or experiment, grammatical or semantic information is used. The next section presents different ways in which to evaluate items once they have been generated.

IV.2 Item analysis

When generating items automatically, one important point is to create good items. With an automatic process, the amount of the generated items is less important than their quality. Some aspects of the quality of the items are automatically detectable. For instance, it is possible to find incorrectly formed distractors. However, we also need to evaluate the item manually in order to obtain results based on real scenarios. This evaluation is based on the item analysis theory.

Item analysis theory reviews items qualitatively and quantitatively, with the aim of identifying problematic items. The qualitative analysis is usually

[21] In cases in which the generation of more distractors is possible, it would be possible to reject only the duplicated candidates.

based on experts' knowledge, whereas the quantitative analysis is conducted after the items have been given to students, i.e., statistical analysis.

IV.2.1 Correctness of the questions

The qualitative analysis of item responses gives us a way to measure the correctness of the automatically generated questions. In order to do so, the questions are analysed by experts. Thus, when the aim was only to measure the correctness of the question, we took into account the acceptance rate of the experts. When more than one expert took part in the evaluation, we used the kappa index[22] to analyse the results. This measure will be used in section V.3.1.2 to evaluate the correctness of the generated declension and verb tests.

Although it is assumed that all of the experts followed the same instructions for the evaluation of the automatically generated questions, we must also consider other aspects, such as chance and some personal factors, which may also have influenced the results obtained in the evaluation. These factors could be: (i) the experts' own experience when generating questions manually; (ii) the end-users of the questions in the experts' minds; (iii) when and how the evaluation was carried out; (iv) the number of questions to evaluate, etc. Cohen's kappa index (κ) (Cohen, 1960) takes these variables into account.

Cohen's kappa coefficient is a statistical measure of inter-rater agreement. It is a more robust measure than the percent agreement because the kappa takes into account the possibility of agreement occurring by chance. The equation for κ is:

$\kappa = \frac{\Pr(a)-\Pr(e)}{1-\Pr(e)}$

where:

Pr(a) = Observed percentage of agreement;

Pr(e) = Expected percentage of agreement.

If the raters are in complete agreement, then $\kappa = 1$. If there is no agreement between the raters (other than what would be expected by chance), then $\kappa <= 0$.

Landis and Koch (1977) classified the values as: <0 = no agreement; 0-0.20 = slight agreement; 0.21-0.40 = fair amount of agreement; 0.41-0.60 =

[22] The kappa index is not a measure of item analysis.

moderate amount of agreement; 0.61-0.80 = substantial amount of agreement and 0.81-1 = almost perfect agreement.

Cohen's kappa measures agreement between two raters. When there are more than two raters, it is possible to use Fleiss' kappa (Fleiss, 1971), a similar measure of agreement.

IV.2.2 Quality of the questions

When the aim was to measure the quality of the automatically generated questions, two kinds of experiment were carried out: (a) experiments in which the questions were evaluated qualitatively and quantitatively; and (b) experiments in which the questions were evaluated only quantitatively.

In a qualitative evaluation, the questions are given first to an expert who has to select the best candidate sentences and distractors from among those which have been generated.

The quantitative analysis of item responses provides descriptions of item characteristics and test score properties, among other things. In this work, we explored item difficulty, item discrimination and the evaluation of distractors based on classical test theory (CTT).

Item difficulty: The difficulty of an item can be described as the proportion of students who answer the item correctly. The higher the difficulty value, the easier the item.

Item discrimination: This index indicates the discriminatory power of an item. That is, an item is effective if those with high scores tend to answer it correctly and those with low scores tend to answer it incorrectly.

The point-biserial correlation is the correlation between the scores that students receive on a given item and their total scores. A large point-biserial value indicates that students with high scores on the overall test got the item right and that students with low scores on the overall test got the item wrong. The point-biserial correlation is a computationally simplified Pearson's r between the dichotomously scored item and the total score. In this approach, we use the corrected point-biserial correlation. That is, the item score was excluded from the total score before computing the correlation. This is important because the inclusion of the item score in the total score can artificially inflate the point-biserial value (due to the correlation between the item score and itself).

There is an interaction between item discrimination and item difficulty. It is necessary to be aware of two principles: very easy or very difficult test items have little discriminatory power, and items of moderate difficulty (60% to 80% of students answer correctly) generally have more discriminatory power. Item difficulty and item discrimination measures are useful only to help to identify problematic items. Poor item statistics should be put down to ineffective distractors.

Distractor evaluation: in order to detect poor distractors, the option-by-option responses of high-scoring and low-scoring student groups will be examined when the results are shown. With this purpose, two kinds of results will be presented: the number of distractors chosen and an explanation.

The results of this analysis are presented in chapter VI, for which some constraints were previously established. The analysis of the item difficulty and item discrimination values for different sets of students led us to set 30 as a reasonable sample-size for the experiments. In this work, we marked an item as easy if more than 90% of students answered it correctly. On the other hand, an item was defined as difficult when less than 30% of the students chose the correct answer. The desired item difficulty value is 0.5. In contrast, the results of item discrimination and the evaluation of distractors were obtained based on the low-scoring and high-scoring students. The top third of students with the highest scores in the given test were considered as the high-scoring group, while the bottom third of the class were considered to be the low-scoring group. Therefore, these three measures are used in chapter VI to explain the analysis of the results of the 951 students.

IV.2.3 Experts' evaluation

Item authoring is one of the essential functions that an item banker must perform (Vale, 2006). Thus, as our system is a plausible way for an item banker to create items, we also offer an item authoring tool. This type of application has to be accessible to the user (editors) in terms of functionality, navigation and speed. In addition, the option of offering the items in two different ways is of interest: as the end-users (students, in our case) will see them, and as an advanced method of editing in which more technical aspects can be consulted or updated.

In order to do so, as well as to acquire experts' knowledge, we have implemented a web-based **post-editing environment**. The post-editing environment requests ArikIturri to generate questions of any type. These items, which are represented by means of the question model, are imported into the environment's database. The importation of the questions implies a match between the concepts defined in the question model and the representation of the domain of the assessment application.

The application offers the option of studying, modifying or correcting the generated questions for multiple-language items. For this reason, the interface is also multilingual. Moreover, the application has two types of user: post-editors and supervisors.

Post-editors

Post-editors are responsible for analysing the automatically generated questions in order to evaluate the quality of the generated items as well as to give feedback to the system. In order to do so, the environment offers the option of adding comments relating to the reasons for not accepting or modifying the questions. Thus, post-editors have different options regarding each generated question: to *accept* it on its own; to *discard* it if it is not an appropriate item; or to *modify* it. Those actions are applicable at two levels: **sentence level** and **distractor level**.

Sentence level
As regards the stems, in the first version of the environment, the interface offered four options:[23] (a) acceptable; (b) acceptable with minor revisions (e.g., punctuation); (c) acceptable with major revisions (e.g., grammar); and (d) unacceptable.

As is explained in chapter V, our experiments showed that it is possible to define the main reasons for discarding stems as:

- The sentence length is inappropriate;
- A larger context is needed;
- The stem is too difficult for learners.

As a consequence, as these reasons are applicable to any language, we added these reasons to the interface, thereby offering a more precise application.

[23] We started from the options offered by Mitkov *et al.* (2006).

Keys
Regarding the keys of the items, post-editors can never modify the original correct answer (the key), but they can add a new one. For us, it is interesting to keep the original answer even if it is replaced by the post-editor. The option of adding new answers allows us, for instance, to increase the difficulty of the item or to study the influence of the occurrences of the key across the test as a whole by replacing the answer with a synonym (cf., section VI.4.3.4). As adding new answers is done through the environment, a new item to be stored in the item bank is created, but ArikIturri does not notice the change because it is carried out after the export process.

Distractor level
As regards distractors, post-editors can update them or add new ones if they consider that other distractors are more appropriate. They can also accept or reject the distractors. When discarding a distractor (e.g., because there is more than one possible correct answer among the options) the post-editor has to determine the reason for doing so. In this way, it is possible to improve the heuristics based on experts' knowledge (cf., section V.3). In order to do so and to offer a way of understanding the automatic process when generating the distractors, the information relating to the heuristics is offered by the application. In addition to offering this information in an accessible way, the application stores the post-editor's opinion of the heuristics. This process is presented in an easy-to-understand way, avoiding any technical information.

Thus, the modifications of the post-editors are used in two ways: on the one hand, to improve the system (giving feedback which is stored in the question model) and, on the other hand to create new questions for the final users, the teachers.

Supervisors

As there may be more than one post-editor, more than one version of the same question could appear. These are the questions which are supervised by the supervisor user. Therefore, the supervisor oversees the items generated by the post-editors and is responsible for selecting the best sentence as well as the distractors.

Although the ideal scenario is to have a two-step evaluation, for those cases in which this is not an affordable option, the supervisor's job is avoid-able.

In Conclusion

In this chapter, we have presented the main resources used by ArikIturri. The system makes use of various types of resource in order to generate items: corpora; ontologies; dictionaries; syntactic analysers and morphological generators. Their features as well as the tasks for which they have been used have been explained. The aim of this analysis has also been to study the influence of the resources used in the generation process, as the quality of those resources can determine the quality of the system and the generated questions. In addition, we have presented different ways to evaluate the automatically generated items. Although it is an expensive task, it is necessary in order to discern the quality of the tests.

CHAPTER V

Using grammar when generating test items

In this chapter, we present the ways in which ArikIturri employs grammatical information in different steps of the generation of items. We have taken advantage of the available Basque language resources and the experiments were conducted during the development of the system. Hence, this chapter devotes its attention to the study of several approaches to the creation of tests.

V.1 Introduction

In this chapter, the experiments relating to the use of grammatical information are presented. The aim is to study the usefulness of this type of linguistic information in different steps of the generation process. Thus, we have tested the modules of the architecture specified in the system's design. As mentioned in sections II.4.3 and IV.1, grammatical information is used by the *sentence retriever* and *answer focus identifier* modules of ArikIturri to identify the topic of an item.

In addition, the *item generator* module can require morphosyntactic information to create the components of the items. The available linguistic resources can play a beneficial role in this generation. Therefore, we have paid special attention to the stem and distractor generation tasks. For this purpose, we first studied the corpora and NLP tools within reach. Based on this analysis, we designed four experiments to investigate the applicability of integrating the use of grammatical information into the creation of items.

As regards the stem generation task, the main difficulty lies in the transformation of the source sentence. The experiment presented in section V.2 addresses this matter by proposing a methodology to integrate the modification process into ArikIturri. For this purpose, we focused on the ZT corpus. The experiment created questions (interrogative stems) regarding numerical entities.

The rest of the experiments are focused on different strategies for generating distractors within the Basque language learning scenario. Two of the experiments create items to deal with declension cases and verb forms and another one is focused on the correct use of determiners. These three topics (declension, verbs and determiners) have been selected based on the accessible resources and experts' experience.

The first approach to defining the heuristics is expounded in section V.3.1. This experiment relies on the simplest strategy, from a computational point of view, for defining heuristics. That is to say, the responsibility is delegated to humans, and thus the quality of the generated distractors should be high. When this criterion is applied, the system has two main duties. One is the detection of the required grammatical information in the candidate sentences. The other is the generation of the corresponding distractors to deal with declension cases and verb forms.

As the defined heuristics aim to test the language level of students, the language learning corpus was considered to be the most suitable corpus for this scenario. Although the heuristics are based on common learner errors, the main disadvantage of this approach relates to the fact that it is an expensive process and dependent upon human generators.

The second experiment presented in section V.3.2 studies two interesting resources that can reduce the cost of defining appropriate heuristics to simulate experts' behaviour. One is the learner corpus and the other is the work done as regards the automatic detection and correction of determiner errors (Uria, 2009). Therefore, the combination of both tools can offer an alternative way to define heuristics based on learners' errors. Nonetheless, the automatic detection of errors is based on manually generated rules which are still effort-intensive.

The last experiment explained in section V.4 proposes a complete automatic methodology for setting the heuristics. With this purpose, the experiment is focused on the work done by Aldezabal *et al.* (2003), in which linguistic patterns are extracted automatically. More specifically, these patterns are a valuable source for automatically acquiring the necessary linguistic information to generate items to work with declension cases and verb forms. Although this pattern extraction task was originally conducted in a general corpus, our aim is to apply this information to items from the Basque language learning corpus.

V.2 Stem generation

A new community of interdisciplinary researchers[1] have found a common interest in generating questions.[2] The QGSTEC (Rus and Graesser, 2009) began the discussion on the fundamental aspects of question generation (QG) and set the stage for future developments in this emerging area. Therefore, QG is defined (Rus and Graesser, 2009) as the task of automatically generating questions from some form of input, for which the input could vary from raw text to in-depth semantic representation.

With this purpose, Rus and Graesser (2009) listed the necessary components for any shared QG task: (1) sources of information; (2) input text; (3) QG system; (4) processing goals; (5) output questions; and (6) the evaluation of questions. In addition, they also state that the questions are generated in accordance with the system's goals and that the quality of the questions is directly dependent upon the extent to which they fulfill these goals.

Among the various tasks which have been proposed, we focus on the text-to-question task, in which the goal is to generate a set of questions for which the given text implies answers. Figure V.1 shows this task as proposed by Rus and Graesser (2009).[3]

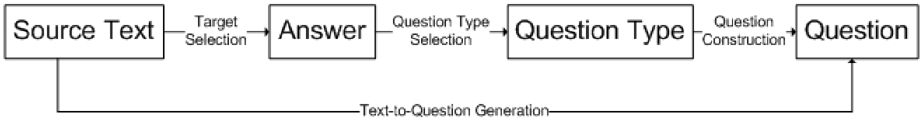


Figure V.1: Text-to-question generation task

[1] Researchers from various disciplines such as cognitive science, computational linguistics, computer science, discourse processing, educational technologies and language generation.

[2] http://www.questiongeneration.org/

[3] These steps correspond with what Nielsen (2008) calls concept selection, question type determination, and question construction.

In brief, first, in the *Target Selection* step, the topic is identified.[4] Next, through the *Question Type Selection* process, the question type is selected. Finally, by means of the *Question Construction* step, the surface form of the question is created based on the previous steps.

As regards the evaluation measures, Rus and Graesser (2009) mention that QG systems can be evaluated either manually or automatically. For a manual evaluation, they propose a panel of human judges. For an automated evaluation, they suggest the creation of tools similar to ROUGE and BLEU,[5] as well as evaluation metrics such as precision, recall and fluidity. One particular way of evaluating QG systems is the QGSTEC (2010). This is a way of promoting the evaluation of different systems and obtaining a dataset. It consists of three tasks: (a) QG from paragraphs; (b) QG from sentences; and (c) an open task. The most successful task was the QG from sentences, which may be due to the fact that it is the simplest task. As our aim is to start with this research line, we also worked from sentences to questions. Section V.2.1 presents the general idea behind our proposal.

In our approach, this QG approach is conceived as part of the item generation process. The stem of an item is selected from the input corpus. The source sentence has to include the topic (*Target Selection* process) and, from this point, ArikIturri can generate the stem. Sometimes, the system only keeps the stem as the source sentence, while at other times, the system transforms the order or deletes some chunks of the sentence. Finally, there are times when the source sentence is transformed into a question clause. This is, in fact, one of the difficulties of generating a correct stem: the transformation of a declarative statement into a question (the *Question Type Selection* and *Question Construction* processes).

V.2.1 Question generation

Our question generator system (Aldabe *et al.*, 2011) must be seen as a subsystem of ArikIturri, as it is integrated within the ArikIturri system. However, like the QG community, the QG task is here proposed as an independent task.

Our QG system must be conceived as a shallow question generator which deals with the Basque language. Although it aims to be a complete system in

[4] This is also referred to as the *Key Concept Identification* task.

[5] Both are machine translation evaluation tools.

which the three different steps are fulfilled[6] (see Figure V.1) in a generalised way, the prototype presented here is focused on some sub-tasks. In this way, the target selection was restricted to numerical entities and the experiment was conducted on the ZT corpus.

Example V.2.1 (Sentence from ZT corpus)

Joan den abenduan argitaratu zuen txostena, eta otsailaren 25a arte, nahi duenak iritzia emateko aukera du.

The report was published last December, and, those who want to do so have the opportunity to express their views until February 25th.

Henceforth, the source sentence from example V.2.1 will be used to explain the various steps of the generation process. More specifically, the explanations focused on the date otsailaren 25a (February 25th).

Section V.2.1.1 explains the *Target Selection* task. In this approach, we have differentiated three sub-tasks: clause identification; numerical entity identification; and finally, candidate selection. Once the candidates are selected and based on the detected numerical entities, the corresponding wh-words are identified during the *Question Type Identification* task. Section V.2.1.2 presents the particularities of our approach, through which WHICH, HOW MANY and WHEN wh-words were identified. Finally, section V.2.1.3 addresses the *Question Generation* task, in which some transformation rules are proposed in order to modify the source information to obtain the corresponding question.

V.2.1.1 Target selection

In this experiment, the target selection task is divided into: (i) the identification of clauses; (ii) the identification of numerical entities; and (iii) the selection of candidates.

Although all of the works presented in QGSTEC (Boyer and Piwek, 2010) deal with the English language and our proposal is focused on the Basque language, the simplification of the input sentences is a matter of study in both scenarios. In fact, an important issue in QG is how to generate concise questions from complex sentences (Heilman and Smith, 2010).

[6] The three steps from Figure V.1 are: *Target Selection*; *Question Type Selection*; and *Question Construction*.

Once the clauses have been identified, the identification of numerical entities is conducted. As a consequence, we have not identified the most appropriate concept forms with which to construct the questions nor the key question-worthy concepts in the knowledge source, as Becker *et al.* (2010) propose.

Step 1: Clause identification

In this approach, the aim is to obtain clauses from the source sentence in order to generate questions. More specifically, in this first experiment, the system selects the coordinated clauses.

In our approach, the identification of clauses is carried out by means of the combination of rule-based grammar with machine learning techniques (Alegria *et al.*, 2008). More specifically, it is based on a learning model that recognises partial syntactic structures in sentences (Carreras *et al.*, 2005) and incorporates features designed to represent syntactic phrases. This property is used by Alegria *et al.* (2008) to include linguistic features, by applying different combinations of the features in order to obtain the best results. Thus, the main idea is to recognise partial syntactic structures in a sentence by means of machine learning techniques.

Carreras *et al.* (2005) proposed a filtering-ranking architecture. In the filtering layer, the boundaries of clauses in the sentence are detected. The ranking layer classifies the candidates and the final solution is computed with a dynamic programming algorithm that builds the best structure of clauses for the sentence.

Alegria *et al.* (2008) applied different combinations of the features in order to obtain the best results. For this purpose, they first set a baseline system which puts clause brackets only around the sentences obtaining a rate of F1 of 37.24%.[7] Initial experiments used information concerning words, PoS, chunks and clauses. After that, they added features such as subcategories, declension information, lemmas, subordinate clauses as well as the information regarding clause splits which is obtained by means of rule-based grammar.[8] Their results show that the more linguistic information they added, the better their results. In addition, they concluded that the addition of rule-based grammatical information improved the results considerably (an improvement

[7] The same baseline system for English achieves a score of 47.71%.

[8] Rule-based grammar was originally used to tag noun and verb chains as well as sentences and clauses.

of two points). Therefore, the clause identifier that used all the mentioned features obtained an F1 of 58.96%. This is in fact the combination used by our QG system.

Once this step was applied, when given the sentence from example V.2.1, the system detected two coordinated clauses: Joan den abenduan argitaratu zuen txostena (The report was published last December) and Otsailaren 25a arte, nahi duenak iritzia emateko aukera du (Those who want to do so have the opportunity to express their views until February 25th).

Heilman and Smith (2010) went one step further and proposed to generating questions not just about the information in the main clause, but also about the information embedded in nested constructions. In our first approach, our system detected these subordinate clauses but rejected them due to the lack of a main verb. However, in the future, we plan to transform these candidates in order to generate questions from them as well.

Step 2: Numerical entity identification and classification

Numbers appear in many different ways in Basque written texts. Due to the fact that Basque is an agglutinative language, even numbers make up different word forms. In addition, numerical entities can express a wide range of information such as percentages, magnitudes, dates, times, etc. Although most numbers follow a simple pattern (digit before the unit of measurement or category), the difficulty lies in some compound structures such as percentages or pairs of numbers with a conjunction between them. In general, patterns in which the category and the number are far away from each other are difficult to treat. Moreover, special attention must be paid to the order of the words in the phrase. Occasionally, the number can appear after the category, e.g., 2 lagun, lagun 2 (two friends,friends two).

Once the clauses are identified, the numerical entities within the clauses are classified based on a Numerical Entity Recogniser and Classifier for Basque (NuERCB) (Soraluze *et al.*, 2011). More specifically, NuERCB decides whether these numbers express a date or time, are associated with units of measurement, or refer to common nouns.

The input used by NuERCB is provided by *Ixati*, which identifies and tags numbers according to six predefined types: ZEN (used to mark nondeclined numbers written with digits); ZEN_DEK (used for declined numbers); HAUL_ZNB (used for multiword numbers); HAUL_DATA (used when a multiword date structure is detected); ERROM (used for Roman numerals); and DET DZH (used for numbers written in characters).

The range of categories addressed by NuERCB is wide. On the one hand, there are categories associated with specific properties such as area, density, length, temperature, time, etc. that are represented by units or symbols: metre (m), kilogram (kg), second (s) etc. These categories are denoted as closed. On the other hand, each common noun or concept can be considered as an open category; for example, in the phrase 20 books, the noun book plays the role of an open category which is linked to the number 20.

In the case of closed categories, the goal is to mark numerical entities along with the property to which they refer and the unit or symbol which is used. For example, in the sentence Hegazkinak 2000 km/h-ko abiaduran mugi daitezke (The aeroplanes can fly at 2000 km/h), 2000 is labelled with two tags: the symbol of measurement is **km/h** and the associated property is **speed**. Authors have also pointed out that determining the boundaries of numerical entities would be necessary in some composed structures like 21 ordu 5 minutu eta 12 segundo (21 hours, five minutes and 12 seconds).

In the case of open categories, they distinguish between percentage expressions like hazkundea % 10ekoa izan da (there has been a 10% growth), and simple numbers or amounts like 1250 biztanle (1250 inhabitants). In these cases, the system determines which common noun refers to the numerical entity: 10% is linked to hazkundea (the growth) and 1250 is linked to biztanle (inhabitants).

NuERCB compiled a set of hand-crafted rules which have been implemented in Finite State Transducers (FST). They have defined 34 FSTs to classify closed categories and two more for open categories that refer to common nouns. The rules were defined using *foma* (Hulden, 2009) and, in total, the set of FSTs is composed of 2095 hand-crafted rules which are able to identify 41 properties, 2006 units and 1986 symbols. According to the MUC evaluation method, NuERCB obtains an F1 score of 86.96% and, in line with Exact-Match scoring, this score reaches 78.82% for the total of the categories.[9].

Based on the two coordinated clauses detected in the *Clause Identification* step, NuERCB detected two numerical entities: abenduan (in December) and Otsailaren 25a (February 25^{th}).

[9] Those are two well-known evaluation methods.

Step 3: Candidates selection

After the numerical entities have been classified and tagged, the candidate clauses have to be identified. At this point, the QG system takes into account those clauses which have at least one tagged number. In addition, once the clauses incorporating the topic have been detected, the verb information is also consulted. In order to be a candidate, the clause has to comprise one and only one main verb. Furthermore, if the candidates are clauses which are part of other clauses, the system considers the shortest candidate clauses only. This step must be carried out because the clause identification task is not perfect, due to the recursive nature of the clause structures.

We have previously mentioned that the aim is to detect two coordinated clauses. However, the source sentence also contains subordinate clauses, as represented in example V.2.2 with parentheses.

Example V.2.2 (Subordinate and coordinate clauses)

((Joan den) abenduan argitaratu zuen txostena) eta (otsailaren 25 arte, (nahi duenak) (iritzia emateko) aukera du)

(The report was published (last) December), and, (until February 25th, (those who want to do so) have the opportunity (to express their views))

The selection of the shortest candidate is performed as the final step of the selection process in order to lose as little numerical information as possible. For instance, based on example V.2.1, the system proposes as candidate clauses, among others, otsailaren 25a arte nahi duenak,[10] otsailaren 25a arte nahi duenak iritzia emateko[11] and otsailaren 25a arte nahi duenak iritzia emateko aukera du.[12]

If the first step was to select the shortest clause, the system would choose otsailaren 25a arte nahi duenak. It is a clause that contains a tagged number, but it does not contain a main verb. This would mean that, in the end, the system would not take any of them into consideration as candidate sentences. In contrast, in this order, the system chooses as a candidate otsailaren 25a arte nahi duenak iritzia emateko aukera du.

[10] Until February 25th those who want to do so

[11] Until February 25th, those who want to do so to express their views

[12] Until February 25th, those who want to do so have the opportunity to express their views

V.2.1.2 Question type identification

	Pattern	wh-word
1	((IZE ZEN or DET DZH))[13]	ZENBAT (HOW MANY)
2	((IZE ZEN or DET DZH)) and ERG[14]	ZENBATEK (HOW MANY)
3	((IZE ZEN or DET DZH)) and DAT	ZENBATI (TO HOW MANY)
4	((IZE ZEN or DET DZH)) and GEN	ZENBATEN (OF HOW MANY)
5	((IZE ZEN or DET DZH)) and SOZ	ZENBATEKIN (WITH HOW MANY)
6	((IZE ZEN or DET DZH)) and DES	ZENBATENTZAT (FOR HOW MANY)
7	((IZE ZEN or DET DZH)) and INS	ZENBATEZ (BY HOW MANY)
8	((IZE ZEN or DET DZH)) and INE	ZENBATETAN (HOW MANY TIMES)
9	((IZE ZEN or DET DZH)) and ABL	ZENBATETATIK (OUT OF HOW MANY)
10	((IZE ZEN or DET DZH)) and ALA	ZENBATE(TA)RA (TO HOW MANY)
11	((IZE ZEN or DET DZH)) and ABU	ZENBATE(TA)RAINO (TO WHICH EXTENT)
12	((IZE ZEN or DET DZH)) and ABZ	ZENBATE(TA)RANTZ (TOWARDS HOW MANY)
13	((IZE ZEN or DET DZH)) and GEL	ZENBATEKO (WHAT AMOUNT)
14	((IZE ZEN or DET DZH)) and ALA+GEL and BIZ[15]	ZENBATERAKO (TO HOW MANY)
15	((IZE ZEN or DET DZH)) and GEN+INE and BIZ	ZENBATENGAN (IN HOW MANY)
16	((IZE ZEN or DET DZH)) and GEN+ABL and BIZ	ZENBATENGANDIK (FROM HOW MANY)
17	((IZE ZEN or DET DZH)) and GEN+ALA and BIZ	ZENBATENGANA (TO HOW MANY)
18	((IZE ZEN or DET DZH)) and GEN+ABZ and BIZ	ZENBATENGANANTZ (TOWARDS HOW MANY)
19	((IZE ZEN or DET DZH)) and GEN+ABU and BIZ	ZENBATENGANAINO (UP TO HOW MANY)
20	((IZE ZEN or DET DZH)) and GEN+MOT and BIZ	ZENBATENGATIK (FOR HOW MANY)
21	MAG_DATA and GEL	NOIZKO (WHEN FOR)
22	MAG_DATA and ABL and ALA	NOIZETIK NOIZERA (WHEN FROM ... TO)
23	MAG_DATA and ABL	NOIZTIK (WHEN FROM)
24	MAG_DATA and ALA	NOIZ ARTE (WHEN UNTIL)
25	MAG_DATA	NOIZ (WHEN)
26	MAG_*	ZENBAT (HOW MANY)
27	DET ORD	ZENBAGARREN (WHICH)
28	...	...
29	DET ORD and INE	ZENBAGARRENEAN (IN WHICH)
30	...	...

Table V.1: Patterns for recognising numerical entities

Once the final candidates are obtained, the QG system is responsible for identifying the corresponding wh-word. Thus far, we have implemented and tested wh-words relating to measures, dates, times and numbers. As with other words, the Basque wh-words also make different word formation. Thus, the system incorporates patterns to recognise first the numerical entities and

[13] IZE: noun; ZEN: number; DET DZH: numbers written in characters.

[14] ERG: ergative; DAT: dative; GEN: genitive; SOZ: sociative; DES: destinative; INS: instrumental; INE: inessive; ALA: allative; ABU: abulative; ABZ: and GEL: locative genitive.

[15] BIZ: animate. ZENBATERAKO, ZENBATENGAN, ZENBATENGANDIK, ZENBATENGANA, ZENBATENGANANTZ, ZENBATENGANAINO and ZENBATENGATIK wh-words asked about animates.

then the morphosyntactic information in order to establish the corresponding wh-words. Table V.1 shows the integrated patterns.

For instance, if the category of the detected numerical entity is a noun (IZE) or a determiner (DET) and the corresponding noun is marked with the dative case, the corresponding wh-word is ZENBATI (TO HOW MANY). In the sentence Sei laguni gertatu zitzaien (It happened to six people), the chunk Sei laguni (to six people) contains an open numerical entity. This entity is tagged and classified by the system, so we know that there is an open numerical entity with the number sei (six) and the corresponding noun lagun (people). In addition, six is a determiner (DET) and a number (DZH) and laguni (to people) takes the dative case. Therefore, the system replaces the number six with the wh-word ZENBATI (TO HOW MANY).

If the phrase containing the numerical entity does not match with any of the other patterns regarding nouns and determiners which are tagged as open categories (from rows 2 to 20 in Table V.1), then the wh-word ZENBAT (HOW MANY) is used. Note that the patterns from 2 to 20 are necessary in order to deal with the different word forms that the numbers can make.

In Table V.1, the patterns from rows 21 to 25 refer to the numerical entities that are related to dates. This type of numerical entity always corresponds to a WHEN wh-word that, depending on the declension case, varies in its form. For instance, the date expression 1990eko abenduko (in December of 1990) needs the time wh-word NOIZKO (WHEN FOR) because it is a date magnitude and the last component of the entity (abenduko - in December) contains the locative genitive mark.

The pattern "MAG_*" refers to all closed numerical entities that are not related to dates. In the case of these closed categories, we decided to generate only the ZENBAT (HOW MANY/MUCH) wh-word, because these closed magnitude entities always have at least two components (the number and the corresponding magnitude) and the number is never marked with a declension case.

The last set of patterns in Table V.1 that contain "DET ORD" have been defined in order to work with ordinal numbers. As occurs with patterns relating to open numerical entities (from row 1 to 20 in Table V.1), the ordinals can also be marked with different word forms. For instance, while the wh-word which corresponds to the numerical entity laugarren postua (the fourth position) is ZENBAGARREN (WHICH), in the case of XI.ean (in the 11th) the corresponding wh-word is ZENBAGARRENEAN (IN WHICH), because the ordinal has the inessive mark. Therefore, as in the case of open numerical entities, 20 patterns have been defined in order to work with ordinals.

Finally, it is necessary to point out that numbers that refer to a percentage value are treated as open numbers and ordinals. The only difference is the addition of the word EHUNEKO (PERCENT) before the generated wh-word.

Based on the defined rules and as regards the previously detected numerical entity Otsailaren 25a (February 25^{th}), the chunk that corresponds to it is Otsailaren 25a arte (until February 25^{th}). It is a date magnitude, and the last component of the date expression contains the allative case. Therefore, the corresponding wh-word is NOIZ ARTE (UNTIL WHEN).

V.2.1.3 Question generation

Once the who-word is set, before constructing the question, some modifications to the source sentence have to be carried out: (i) in the event that the main verb is in the first singular or plural person, the tense is transformed into the corresponding third person; (ii) linking words used to connect sentences are deleted from the sentence; (iii) in the event that there is more than one numerical entity in a sentence, we only consider the one that is closest to the verb on its left; (iv) if all of the entities appear on the right-hand, we also mark the closest to the verb; and (v) finally, the system constructs the question.

The question building is based on some simple transformation rules defined in the system. First, the generated wh-word followed by the rest of the words of the chunk in which the numerical entity is located is set as the beginning of the question. Following, the main verb is established. After the main verb, the rest of the chunks that are to the right of the verb are included. Finally, the chunks that appear on the left are added. Coming back to example V.2.1, the system generates the question displayed in example V.2.3.

Example V.2.3 (Question generated from source sentence)

NOIZ ARTE du nahi duenak iritzia emateko aukera?
UNTIL WHEN do those who want to do so have the opportunity to express their views?

V.2.2 Evaluation

The evaluation method proposed by Boyer and Piwek (2010) defines some guidelines for human judges. They set five criteria: relevance; question type; syntactic correctness and fluency; ambiguity and variety. The relevance measure takes into account how relevant the questions are to the input sentence. The question type measure indicates that questions have to be of the specified target question type.[16] The syntactic correctness and fluency criterion classifies the built questions according to their syntactic correctness, while ambiguity ranks questions according to their ambiguity grade. Finally, the variety measure is defined to see how different the questions are from each other.

In our QG system's evaluation, we focused on the syntactic correctness and fluency criterion. For this criterion, our human judge followed the same classification as proposed in Boyer and Piwek (2010), and we added some specifications regarding the grade of changes. Table V.2 shows this scoring. For instance, we specified that when a question is grammatically correct and idiomatic (rank 1), there is no need to change any of its components.

Rank	**Description**	**Changes**
1	The question is grammatically correct and idiomatic/natural	No changes
2	The question is grammatically correct but does not read fluently	Minor change
3	There are some grammatical errors in the question	Major changes
4	The question is grammatically unacceptable	Discard

Table V.2: Scoring for syntactic correctness and fluency

Based on the low agreement results obtained in the QGSTEC (Yao, 2010), this evaluation was carried out by one human rater and Table V.3 summarises the results obtained. The results show that 39.34% of the evaluated questions are grammatically correct and do not need any changes (rank 1), while 22.95% are also grammatically correct but need some minor changes. Thus, 62.29% of the questions can be considered to be grammatically correct, while 9.83% of the questions contained some major errors which meant that there was a real need to revise them. Finally, 27.86% of the evaluated questions were discarded.

[16] In the given data, for each sentence, they provide the question types that can be generated.

Rank	#Questions
No changes	24 (39.34%)
Minor	14 (22.95%)
Major	6 (9.83%)
Discard	17 (27.86%)

Table V.3: Manual evaluation results

In addition, we also studied the question type asking one expert to judge whether or not the generated wh-words asked about the source sentence. Furthermore, the expert also had to establish whether the question generated by the system would provide an answer relating to the source sentence. In total, 85.24% of wh-words corresponded to the source sentence and 88.52% of the generated questions were related to the source sentence.

In addition to the manual evaluation, the system's performance was determined by precision and recall measures. These measures have also been used in the QGSTEC by some of the authors.

$$precision = \frac{correct}{correct + incorrect}$$

$$recall = \frac{correct}{correct + missed}$$

The precision measure expresses the number of correct numerical entities among those which were detected, while recall shows the number of correct numerical entities out of all of the instances that are actually part of the source. Although these measures are somehow related to the performance of NuERCB, we consider it interesting to calculate them because obtaining the clauses automatically could also influence the results. The system obtained a 84.25% precision level and a 78.26% rate of recall.

Soraluze *et al.* (2011) detected some common structures in Basque like 700 bat km (about 700 km), in which bat corresponds to about. In addition, the word bat can also mean one. As bat is nearer than 700 from the unit of measurement (km), the system's rules would erroneously tag bat as a number. In order to avoid this type of mismatch, our question generator does not consider the numerical entities containing the word bat as candidates.

From the analysis of the generated questions, we detected some minor changes to the system which would improve the generation process. First, months that are written in characters need to be dealt with separately by our algorithm. Second, imperative sentences have to be discarded as candidate sentences. Finally, it is possible to delete adverbs that appear at the beginning of sentences before generating the questions.

We are studying how to improve the use of the previously analysed temporal information, because some information is still being lost. For instance, if a period of time is followed by a word, it is correctly tagged and detected. However, if the period of time comes in brackets and without any corresponding word, the system does not always provide the corresponding wh-word.

This emerging area is of significant interest to our research group and it is an ongoing process.

To sum up, this section has presented an approach in which grammatical information is used. However, the linguistic information contained within the words is useful, not only in the sentence selection process but also when generating distractors. Sections V.3 and V.4 deal with this matter.

V.3 Distractor generation: handmade heuristics

Grammar tests are undoubtedly one way of measuring the acquired knowledge of language learners studying the Basque curricula. Thus, offering teachers automatically generated items can be beneficial. On the one hand, the time taken to prepare the tests would be reduced, as teachers would not need to create tests from scratch nor look for resources. On the other hand, items based on up-to-date texts could be more real and interesting for students. Thus, it makes sense to ask ArikIturri to generate this type of item. Obviously, the use of grammatical information within the generation of this type of item is vital.

In general, the linguistic information needed for the automatic generation of distractors can be obtained from different resources such as corpora, ontologies and so on. Moreover, its acquisition can also be carried out in different ways: asking experts in the field, automatically and so forth.

This section is devoted to the automatic generation of distractors in a Basque language learning scenario. More specifically, we focus on the use of manually defined heuristics to address determiners, declension and verbs. All of the defined heuristics aim to test the language level of students and they are defined taking into account the errors made by learners.

Section V.3.1 presents the experiments relating to declension cases and verb forms. For this purpose, the system considers as the input corpus the language learning corpus. The definition of the heuristics to deal with these two topics is assigned to experts in the field.

Section V.3.2 focuses on the correct use of determiners. This experiment studies two available resources in order to simulate experts' behaviour. In this way, the generation of items is based on the Basque learner corpus and the work done by Uria (2009) as regards the automatic detection of errors.

V.3.1 Declension and verb tests

As previously mentioned, in the experiments presented in this dissertation, the system generates grammar tests, and more specifically, tests regarding the correct use of Basque declension cases and verb forms. For this purpose, experts established some common mistakes made by learners when learning the language as the basis of the heuristics. Section V.3.1.1 presents the heuristics in detail and the evaluation results are shown in section V.3.1.2.

As the aim is the generation of distractors, the question types analysed in this section are error correction and MCQ. Example V.3.1 shows an error correction question which deals with the correct use of declension cases.

Example V.3.1 (Error correction example — Declension)

Badaude beldurrari zerikusia duten barreak ere
(There are also some kinds of laughter which have to do to fear.)

V.3.1.1 Heuristics

Before the heuristics can be set by an expert in the field, the source corpus for generating the items had to be chosen. From all of the available Basque corpora (cf., section IV.1.2), the Basque language learning corpus was selected as being the most appropriate in this language learning scenario.

As stated in section IV.1.4, using the Basque language learning corpus as the basis for our items determines the topics which can be addressed. An additional restriction was added to avoid the rejection of some items due to

Declension cases	Change of finiteness	Sociative
		Inessive
	Replacement of declension cases	SOZ => ABS
		SOZ => DAT
		SOZ => ERG
		INE => ABS
		INE => DAT
		INE => ERG
		ABS => SOZ
		ABS => INE
		DAT => SOZ
		DAT => INE
		ERG => SOZ
		ERG => INE
Verb	Change of the person of the verb	DA paradigm
		DU paradigm
		ZAIO paradigm

Table V.4: Heuristics

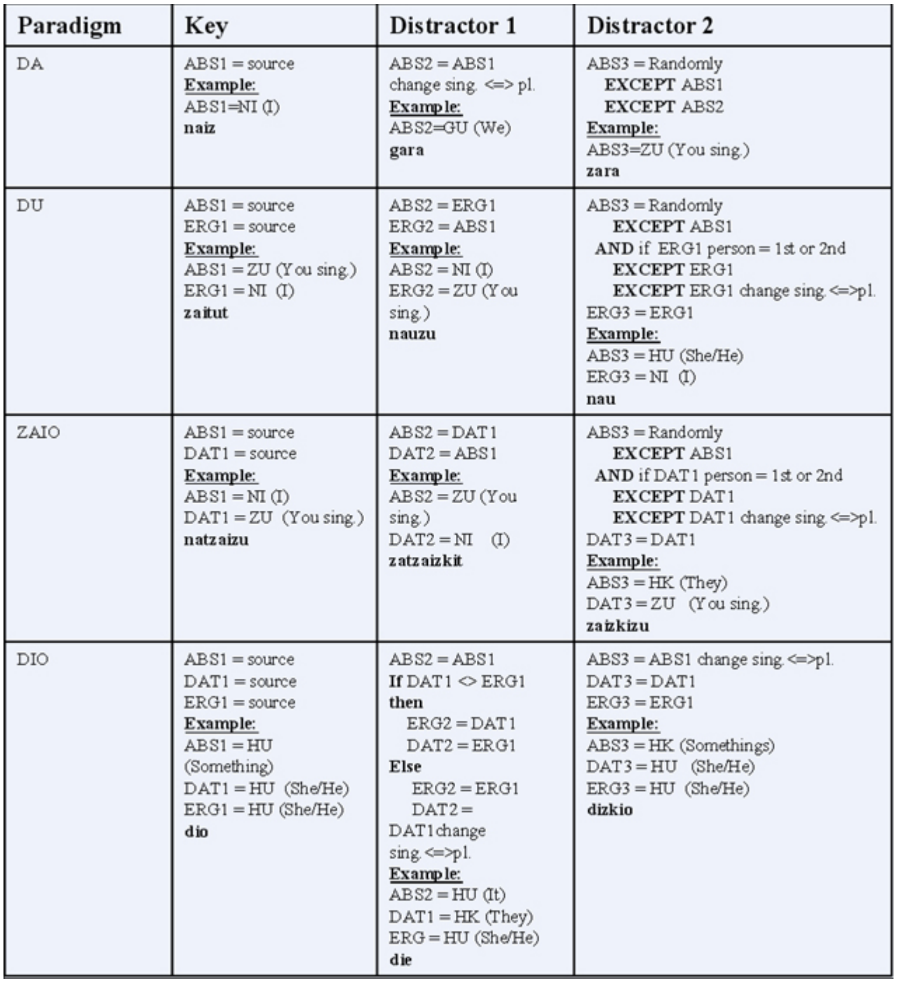
the language level. The source data came from the high language level corpus so that the items would be appropriate for C1 level students (cf., section II.3). After studying the number of appearances of the declension cases and verb forms in the aforementioned corpus, the generation was restricted to five inflection cases and two verb forms: the sociative, inessive, absolutive, dative and ergative cases and present and past indicative verb tenses.

Once the expert's duty was restricted to the high language level and to some specific linguistic phenomena, the expert defined the heuristics in a row. For the declension cases, the common mistakes that students make when learning Basque such as the incorrect use of declension cases or finiteness were taken into account. As regards verb tenses, the heuristics change the different persons of the verb that belong to different auxiliary paradigms. Table V.4 shows the different heuristics and Table V.5 shows how these heuristics are represented in the question model.[17]

[17] Both tables show the same information. Table V.5 can be seen as a generalisation of Table V.4.

Type	Function
declension	change_finiteness(language,case)
declension	replacement(language,source_case,generated_case)
verb	change_person(language,verb_paradigm)

Table V.5: The representation of heuristics in the question model



Paradigm	Key	Distractor 1	Distractor 2

DA	ABS1 = source
	Example:
	ABS1=NI (I)
	naiz
		ABS2 = ABS1
		change sing. <=> pl.
		Example:
		ABS2=GU (We)
		gara
			ABS3 = Randomly
			EXCEPT ABS1
			EXCEPT ABS2
			Example:
			ABS3=ZU (You sing.)
			zara

DU	ABS1 = source
	ERG1 = source
	Example:
	ABS1 = ZU (You sing.)
	ERG1 = NI (I)
	zaitut
		ABS2 = ERG1
		ERG2 = ABS1
		Example:
		ABS2 = NI (I)
		ERG2 = ZU (You sing.)
		nauzu
			ABS3 = Randomly
			EXCEPT ABS1
			AND if ERG1 person = 1st or 2nd
			EXCEPT ERG1
			EXCEPT ERG1 change sing. <=>pl.
			ERG3 = ERG1
			Example:
			ABS3 = HU (She/He)
			ERG3 = NI (I)
			nau

ZAIO	ABS1 = source
	DAT1 = source
	Example:
	ABS1 = NI (I)
	DAT1 = ZU (You sing.)
	natzaizu
		ABS2 = DAT1
		DAT2 = ABS1
		Example:
		ABS2 = ZU (You sing.)
		DAT2 = NI (I)
		zatzaizkit
			ABS3 = Randomly
			EXCEPT ABS1
			AND if DAT1 person = 1st or 2nd
			EXCEPT DAT1
			EXCEPT DAT1 change sing. <=>pl.
			DAT3 = DAT1
			Example:
			ABS3 = HK (They)
			DAT3 = ZU (You sing.)
			zaizkizu

DIO	ABS1 = source
	DAT1 = source
	ERG1 = source
	Example:
	ABS1 = HU (Something)
	DAT1 = HU (She/He)
	ERG1 = HU (She/He)
	dio
		ABS2 = ABS1
		If DAT1 <> ERG1
		then
		ERG2 = DAT1
		DAT2 = ERG1
		Else
		ERG2 = ERG1
		DAT2 = DAT1change sing. <=>pl.
		Example:
		ABS2 = HU (It)
		DAT1 = HK (They)
		ERG = HU (She/He)
		die
			ABS3 = ABS1 change sing. <=>pl.
			DAT3 = DAT1
			ERG3 = ERG1
			Example:
			ABS3 = HK (Somethings)
			DAT3 = HU (She/He)
			ERG3 = HU (She/He)
			dizkio

Figure V.2: Complete specification of the heuristics relating to the verbs[18]

[18] ABS: Absolutive; ERG: Ergative; DAT: Dative. The changing to create the first

With regard to declension cases, there are two general heuristics: the heuristic that changes the finiteness of the key (*change_finiteness*) and the heuristic that replaces the declension case of the key (*replacement*). The expert considered that the most common mistakes regarding the finiteness are related to the sociative and inessive cases, while their replacement is applicable to the five cases which we treated. In addition, the expert also established which cases could be replaced by which others. For instance, if the key is marked with the dative case, the corresponding distractors can be marked with the sociative or inessive cases only. As regards the grammar of verbs, the heuristic is focused on changing the person of the verb (*change_person*). This is carried out with three different auxiliary paradigms: DA, DU and ZAIO (see Appendix A) for the present and past indicative verb tenses. Figure V.2 shows the complete specification of the heuristics comprising the change_person representation. For instance, if the key is natzaizu, a verb form of the ZAIO paradigm of which the absolutive is the first person singular and the dative is the second person singular, the heuristic switches the person of the subject (the absolutive) and the indirect object (the dative) obtaining the candidate distractor zatzaizkit, the absolutive of which is the second person singular and the dative of which is the first person singular.

Although the defined heuristics are language-dependent, the corresponding functions have been defined as generally as possible. That is why, in addition to the declension cases or verb paradigms in question, all of the functions contain a parameter to specify the source language. Example V.3.2 shows the representation of an instance of the replacement heuristic in the question model when generating one specific distractor. More specifically, the absolutive case is replaced by the sociative case, making use of the replacement function.

Example V.3.2

```
<heuristic>
  <type>declension</type>
  <function>replacement(basque,abs,soz)</function>
  <input> blokeoa </input>
  <output> blokeoarekin </output>
</heuristic>
```

distractor in the case of DU and ZAIO paradigms will be carried out only if the persons are different. Otherwise, the candidate will be obtain randomly.

Using these heuristics ArikIturri generated items to be evaluated. In the following, the results of this evaluation are presented.

V.3.1.2 Evaluation

In this section, we present the evaluation which was carried out regarding the correctness of the questions. The basis for all of the experiments presented herein is the 10,079 sentences of the high language level of the Basque language learning corpus which contain at least one topic which addresses one of the five inflection cases or one of the two verb forms. Table V.6 shows the number of instances of each selected inflection case and verb form in the sample corpus.

Topic	**#Instances**
Sociative	703
Inessive	4152
Dative	1148
Absolutive	7884
Ergative	2221
Present indicative	4040
Past indicative	100

Table V.6: Number of instances of each selected topic

Although the system generates four different question types, this evaluation was carried out taking into account the MCQ and error correction question types for different topics. Table V.7 summarises all of the characteristics of the generated items. The error correction questions were generated in order to address the correct use of the inflection cases. More specifically, the heuristic that replaces the declension case was used. The finiteness of the inflection cases was changed when generating MCQs. MCQs were also generated for verbs. Depending on the heuristic which was employed, the generated MCQs had two (in the case of verbs) or three distractors (when dealing with the sociative and the inessive). In total, 17 heuristics were applied (each row of Table V.7 represents a different heuristic) in order to generate the items. For each implemented heuristic, the *candidate selector* module (cf., section II.4.3) selected 100 sentences at random from the corpus

Heuristic		Question type	#Distractors
Finiteness	SOC	Multiple-choice	3
	INE		
Person	DA paradigm	Multiple-choice	2
	DU paradigm		
	ZAIO paradigm		
Replacement	SOZ => ABS	Error correction	1
	SOZ => DAT		
	SOZ => ERG		
	INE => ABS		
	INE => DAT		
	INE => ERG		
	ABS => SOZ		
	ABS => INE		
	DAT => SOZ		
	DAT => INE		
	ERG => SOZ		
	ERG => INE		

Table V.7: The heuristics by question type

once the topics were identified. Thus, the system took into account 1700 sentences.

In chapter II, we mentioned that ArikIturri includes a module for rejecting items. Although the criteria which were defined were not highly sophisticated, the system detected some problematic items. The *ill-formed question rejecter* module discarded 58 MCQs and 292 error correction instances out of the 1700 generated questions. The main reasons for discarding the items were the duplication of distractors (the same word forms but different morphological characteristics) and ill-formed distractors. In this way, 1350 item instances were ready to be evaluated manually: 980 error correction questions and 442 MCQs.

First of all, all 1350 questions were given to one expert in order to measure the acceptance rate and, due to the analysis of the results, some hypotheses were put forward (cf., 1st experiment). Therefore, a sample of the same questions was given to three new experts in order to corroborate the hypotheses and to see the extent to which the good results obtained regarding the acceptance rate were conclusive. In this new analysis, an agreement between the editors was obtained (cf., 2nd experiment). Finally, one last evaluation with new questions was conducted which not only focused on the acceptance rate but also on the evaluation criteria themselves (cf., 3rd experiment).

First experiment: Correctness of the questions

In this first manual evaluation, one expert focused on the correctness of the questions, for which the post-editing environment presented in section IV.2.3 was used. The environment offers different options: to accept the item on its own; to discard it if the question is not appropriate; or to modify it if the editor considers that there is more than one possible correct answer among the options. In this particular evaluation, we asked the expert to modify or reject questions only if they were not well-formed. This set of rejected and modified items gave us a way of evaluating the automatically generated questions.

The expert spent 15 hours evaluating the 908 error correction questions and 442 MCQs. If we believe that all of the questions discarded or modified by the evaluator were not well generated, the results show that the percentage of accepted questions was 83.26% in the case of error correction questions and 82.71% in the case of MCQs. These percentages show us that the automatic generator obtains good results. This assertion becomes even more important if we consider the time that the expert teacher would take to set the questions. It is clear that the setting of the same number of questions with a manual assessment application would be more expensive and time-consuming.

Looking at the results in more detail (see Table V.8) and considering that the number of distractors is higher for MCQs, the percentage of well-formed questions should be higher for error correction questions.

The results obtained in the evaluation confirmed this assumption when dealing with the same topic. In the case of declension cases, the acceptance rate is 82.71% for error correction questions and 64.70% for MCQs. If we analyse the results of the acceptance rate of MCQs, taking into account the number of distractors, there is also a significant difference. The acceptance rate for MCQs containing two distractors is 92.73%, while for those including three distractors it is 64.70%. ArikIturri generates two distractors when dealing with verb tenses and three when dealing with declension cases. In this case, the probability of creating a correct question for verbs is higher

	Topic	Number of distractors	Acceptance rate (%)
Error correction	Declension cases	1	82.71
Multiple-choice	Declension cases AND verb tenses	2-3	83.26
	Declension cases	3	64.70
	Verb tenses	2	92.73

Table V.8: Accepted questions

compared to declension cases. Thus, when generating one distractor (error correction questions), the acceptance rate should be even higher than 92.73%, but the results show a lower acceptance rate.

The methods used for generating distractors and the linguistic phenomena seem to influence the correctness of the questions. On the one hand, the number of distractors changes the acceptance rate of the generated questions. On the other hand, the topic may also have an influence on the results.

Second experiment: Editors' agreement

The aim of this new experiment was to analyse the aforementioned ideas in greater depth. The objective of this new evaluation with more experts was to corroborate the previous results and hypotheses about the topic.

Due to time restrictions, we selected a sample of the items which were evaluated in the previous experiment. The sample contains a total of 431 questions; nearly the same amount for each linguistic phenomenon. Thus, 195 MCQs for verb tenses and 236 error correction questions for declension cases were evaluated by three new experts. Two of them were asked to evaluate the questions which were accepted in the first experiment. The first one had to evaluate the questions relating to verb tenses, the second one answered those relating to declension cases, and the third editor revised the questions which were previously rejected.

Table V.9 shows the results obtained in the second experiment in comparison with those obtained in the first experiment. The editors of the second experiment accepted 94.97% of the questions relating to the verb tenses which were previously accepted in the first experiment and 96.94% of the declension cases. In total, 75% of the questions relating to verb tenses which were rejected in the first experiment were also not accepted in the second one, while in the case of declension cases, this percentage was 25%. More detailed information is given in Table V.10, in which the number of questions on which different editors agreed and disagreed is displayed.

	Verb (%)	Declension cases (%)
Accepted in the 1st experiment	94.97	96.94
Rejected in the 1st experiment	75.00	25.00

Table V.9: Evaluation of the 431 questions

Declension cases		
	Accepted in 2nd experiment	Rejected in 2nd experiment
Accepted in the 1st experiment	190	6
Rejected in the 1st experiment	30	10
Verb tenses		
	Accepted in 2nd experiment	Rejected in 2nd experiment
Accepted in the 1st experiment	170	9
Rejected in the 1st experiment	4	12

Table V.10: Comparison of the results of the two experiments

Both tables show good results; in fact, the second experiment also verifies the high percentage of well-formed questions. The favourable opinion of all of the editors regarding the items is also important as the questions were automatically generated. We applied the kappa measure to our results, and obtained the kappa indices displayed in Table V.11.

	Kappa
Declension cases Error correction	0.28
Verb tenses Multiple-choice	0.61
Total	0.41

Table V.11: Editors' agreement (kappa)

As explained in section IV.2.1, Cohen's kappa (Cohen, 1960) measures the level of agreement between two raters. Based on the classification provided by Landis and Koch (1977), the total agreement between the experts (0.41) was considered to be moderate. However, if we split the results by question type and topic, while the agreement between experts is substantial in the case of MCQs for verb tenses, it is fair in the case of error correction questions for declension cases.

If we take into account that there are more distractors in an MCQ than in an error correction question, the probability that two editors will agree should be higher in the case of error correction questions. In the case of MCQs, they must agree on all of the different distractors. Therefore, as the number of questions evaluated for each question type is almost the same, we should expect better kappa indices in the case of declension cases, as they belong to the error correction type of question. These results are another signal that the topic of the question could influence the results.

Third experiment: Evaluation and generation criteria of experts

By means of this new experiment, we asked the experts to further explain their actions. Although the experts' knowledge was used to establish the heuristics, their experience may also be beneficial in improving the generation process itself. If experts explain their reasons for accepting, discarding or modifying a stem or a distractor, we could try to take advantage of this information.

The main objective as regards the evaluation of the generated items in this new experiment was twofold. On the one hand, as the extraction of the candidate sentences is an automatic process, we wanted to identify what kinds of change the post-editors proposed for the candidate sentences when setting questions about specific topics. On the other hand, we intended to discover the reasons for discarding or updating the automatically generated distractors of MCQs in an attempt to improve the quality of the generation process.

Due to the fact that two experts from HABE were available to work for eight hours each, we adjusted the number of items to be evaluated based on this restriction. The aim was to work with MCQs relating to verbs. Thus, in addition to the 177 previously generated (and evaluated) MCQs designed to deal with the present indicative,[19] 215 new items were generated, leading to a total of 392 MCQs which dealt with verbs.

a) Appropriateness of the stems

Given information	**#Stems**	**Discard (%)**
Stem+Key+Distractors	392	2.55
Stem+Key	205	20.97

Table V.12: Percentage of discarded stems

First of all, we measured the appropriateness of the given stems. As we presumed that the option of consulting the distractors could influence the results, different information was given to the two editors. One of the editors could consult the source sentence, the key and the automatically generated distractors. In contrast, the other post-editor had to analyse the quality of the stems without having the chance to consult any of the candidate distractors. That is, the second post-editor could consult only the stem and the key. Table V.12 presents the results regarding the discard rate of the stems that confirms our first assumption. While in the first case, the percentage of discarded stems is fairly low (2.55%), the number of discarded stems is higher when no candidate distractors are provided (20.97%), constituting a

[19] Comprised of 86 instances of the DA auxiliary type and 91 instances of the DU auxiliary type.

significant difference.[20] Thus, offering candidate distractors within the evaluation task gave editors more detailed information that appears to have helped them to focus on the topic to be analysed in a more restricted way.

As previously mentioned, as well as accepting, discarding or updating the items, the experts also explained their actions. We analysed the collected information and formed some general ideas. Different reasons for discarding the stems were detected. On the one hand, experts found some stems with more than one correct answer, and they considered it to be more convenient to discard these sentences than to change the distractors. On the other hand, some of the sentences were difficult to understand. Sometimes, the post-editors needed more of the context of the sentence in order to understand the topic of the question. In other cases, the ellipses in some of the phrases of the sentence made it difficult to identify the correct form to fill the blank. Finally, it is important to underline that only one sentence was discarded because the blank in the question did not correspond to the selected topic, i.e., the verb.

When updating the stems, different reasons also arose. The post-editors cut sentences that they considered to be too long. They also made changes relating to stylistics aspects or when incorrect aspects of the sentence did not fit the standard definition of Basque grammar. This is an important aspect, as the normalisation process of Basque is currently in progress. The position of the blank in the question was also a reason for updating the stem, and they specifically proposed changing the position of the blank if it was at the beginning of the question.

This analysis led us to make some decisions. After this evaluation, Arik-Iturri no longer allowed the generation of items with a blank in the first position. In addition to this new feature, we identified some reasons for discarding stems that can be extrapolated to any language. Therefore, we add the following reasons for discarding stems to the post-editing environment:[21]

- Inappropriateness of the sentence length;
- The need for a larger context;
- The stem is too difficult for learners.

[20] As the evaluation of the stems with less information is more difficult, fewer items were evaluated.

[21] We also mentioned these features in section IV.2.3.

b) Evaluation of the distractors

When the editors made a decision regarding one item, they had to specify whether the action was motivated because of the stem or because of the distractors. In this way, we also collected data regarding the distractors.[22] In this particular experiment, the results were fairly good, as only 2.05% of the generated questions were discarded due to the distractors. Therefore, among the rest of the questions, 91.83% were accepted and 6.12% were updated for various reasons. Specifically, one of the main reasons for discarding and updating the generated distractors is due to the fact that ArikIturri produced some candidate distractors that could be correct answers. In-depth research studies of these results (the 6.12% and the 2.04%) will give us hints as how to improve the heuristics of the generator.

In conclusion, based on these results, we can conclude that experts' knowledge can be used in different ways: asking them to establish heuristics or picking up the reasons for accepting or discarding the automatically generated items. This knowledge has been used to improve the generation process itself and the post-editing environment. However, we do believe that there is at least one more interesting way of using their knowledge: asking them to generate distractors in a particular scenario and trying to discern their evaluation criteria.

c) Analysis of handmade distractors

With the aim of exploring these criteria, the editor who did not have the opportunity to consult the candidate distractors during the evaluation of the stems was asked to generate distractors. The only established restriction was that the editor had to focus on the grammar of verbs. More specifically, the expert had to work with the present and past indicative and present conditional verb tenses as well as with the DA, DU, DIO and ZAIO auxiliary paradigms (cf., Appendix A). Although in some cases the number of instances of these topics is not high in the source corpus, we decided to extend the topic in order to obtain more results from the editor.

The end-users (the students) were once again C1-level students, and this is why the editor decided to create items that were as complex as possible. We did not restrict the number of distractors per item, and the expert generated 476 distractors for the given 173 stems (2.75 per stem on average). This

[22] The quality of MCQs also depends on the quality of the generated distractors.

proves that the option of building MCQs with three distractors makes sense. Table V.13 displays this information in more detail.

Paradigm	#Items	#Distractors	Average
DA	76	207	2.72
DU	23	69	3.00
ZAIO	66	177	2.68
DIO	8	23	2.88
	173	476	2.75

Table V.13: Number of generated distractors per auxiliary type

The heuristics presented in Figure V.2 do not take into account the option of changing the verb paradigm of the key. In order to see whether the editor considered this feature as a criterion when generating the items, we first looked at the behaviour of the editor regarding this feature. Table V.14 summarises this information.

Paradigm	Keep	Change paradigm			
		DA	DU	ZAIO	DIO
DA	170 (82.12%)	-	31 (14.98%)	5 (2.42%)	1 (0.48%)
DU	147 (83.06%)	23 (12.99%)	-	3 (1.69%)	4 (2.26%)
ZAIO	48 (69.56%)	13 (18.84%)	3 (4.35%)	-	5 (7.25%)
DIO	15 (65.22%)	2 (8.70%)	6 (26.08%)	0	-
	380 (79.83%)	96 (20.17%)			

Table V.14: Number of distractors per verb paradigm and changes

Although changing of the paradigm is not standard practice, the editor changed the paradigm of one candidate distractor per item on average. In addition, there was not variation based on the paradigm of the key. In conclusion, we decided that our system should include this feature when generating this type of items.

We looked not only at changing the paradigm but also at changing the person. This is, in fact, what the previously defined heuristics do. Tables V.15, V.16, V.17 and V.18 present for each verb paradigm the manually generated distractors based on changes to the person.

The heuristic defined in the previous experiments (cf., Figure V.2) to deal with the grammar of the verbs tends to switch between persons in the

		Key — Absolutive	
		Keep	**Change**
Paradigm of the distractor	DA	106 (62.35%)	64 (37.65%)
	DU	20 (64.52%)	11 (35.48%)
	ZAIO	3 (60.00%)	2 (40.00%)
	DIO	1 (100.00%)	0

Table V.15: Changes in the person when the key is absolutive

		Key			
		Absolutive		**Ergative**	
		Keep	**Change**	**Keep**	**Change**
Paradigm of the distractor	DA	14 (60.87%)	9 (39.13%)	-	-
	DU	88 (59.86%)	59 (40.14%)	103 (70.07%)	44 (29.93%)
	ZAIO	3 (100.00%)	0	-	-
	DIO	4 (100.00%)	0	4 (100.00%)	0

Table V.16: Changes in the person when the key is absolutive and ergative

		Key			
		Absolutive		**Dative**	
		Keep	**Change**	**Keep**	**Change**
Paradigm of the distractor	DA	8 (61.54%)	5 (38.46%)	-	-
	DU	3 (100.00%)	0	-	-
	ZAIO	26 (54.17%)	22 (45.83%)	36 (75.00%)	12 (25.00%)
	DIO	5 (100.00%)	0	3 (60.00%)	2 (40.00%)

Table V.17: Changes in the person when the key is absolutive and dative

		Key					
		Absolutive		**Ergative**		**Dative**	
		Keep	**Change**	**Keep**	**Change**	**Keep**	**Change**
Paradigm of the distractor	DA	1 (50.00%)	1 (50.00%)	-	-	-	-
	DU	4 (66.67%)	2 (33.33%)	2 (33.33%)	4 (66.67%)	-	-
	ZAIO	0	0	0	0	0	0
	DIO	11 (73.33%)	4 (26.67%)	11 (73.33%)	4 (26.67%)	10 (66.67%)	5 (33.33%)

Table V.18: Changes in the person when the key is absolutive, ergative and dative

absolutive, ergative and dative cases. In order to see whether the editor's candidate behaved in such a way, we studied the candidate distractors from this point of view. However, we did not find any clear evidence of such a behaviour from the editor.

Although, in the first step, we foresaw only the option of changing the person or the paradigm of the key, the editor's actions gave us new ways of generating distractors. Table V.19 summarises these new options.

Paradigm	**Tense**	**Root**	**Add**	**Delete**
DA	39 (18.84%)	48 (23.19%)	97 (46.86%)	58 (28.02%)
DU	26 (14.69%)	45 (25.42%)	65 (36.72%)	26 (14.69%)
ZAIO	2 (2.90%)	16 (23.19%)	24 (34.78%)	13 (18.84%)
DIO	0	2 (8.70%)	11 (47.83%)	2 (8.70%)
	67 (14.08%)	111 (23.32%)	197 (41.39%)	99 (20.80%)

Table V.19: New ways of generating distractors

Based on the results presented in Table V.19, it seems that in addition to changing the paradigm and person, the addition of some new elements such as the subordinating verbal prefix "bait-" should also be considered by our system when generating distractors. For instance, given the source sentence ... Arellano baino handixeagoa zen Aberin herrian... (... in the town Aberi which was bigger than Arellano ... and the key zen (was), one of the candidate distractors proposed by the expert was baitzen (because was).

In addition, changing the root or tense of the verb or deleting some elements should be taken into account. For example, based on the source sentence Nik ez nuke sekula tatuajerik egingo. (I would not ever have a tattoo.), the expert considered it of interest to delete the negation of the verb as one of the options of the MCQ.

Thanks to this last set of experiments, ArikIturri and the post-editing environment were both improved, enriching them with more information acquired from the experts. In addition, the manual generation of distractors which was carried out by one of the experts give us hints as to how to define new heuristics.

In conclusion, it is clear that the help of experts is useful. However, this way of acquiring knowledge to be integrated into ArikIturri is somewhat expensive. This is one of the reasons for exploring alternative ways to generate distractors. Section V.3.2 presents the experiments based on learners' errors detected in learners' corpora, while section V.4 focuses on the generation of distractors from automatically extracted patterns.

V.3.2 Determiner tests

As previously mentioned, the experience acquired by teachers when teaching a language is a valuable source when establishing the criteria for defining heuristics. In a language learning scenario, when testing students' grasp of grammar, teachers establish distractors based on (among other things) some mistakes which are commonly made by learners. A learner corpus contains a collection of errors made by learners, meaning that this type of corpus is an alternative way of defining the heuristics required to generate distractors.

The learner corpus which was available contains manually tagged determiner errors. Thus, this experiment was based on determiner errors and the work done by Uria (2009). Determiner errors are relatively common in written Basque, specially in learner corpora due to the fact that the use of determiners involves morphosyntactic variation and language learners often tend to confuse Basque and Spanish determiners.[23]

The classification of determiner errors consists of seven main categories (Uria, 2009): deletion of the determiner when it is necessary (D_DET); addition of a determiner when it is unnecessary (A_DET); repetition of the determiner in the determiner phrase (DP) (R_DET); wrong order of the determiner (WO_DET); use of the wrong determiner (W_DET); definite/indefinite names after certain determiners, when they should be indefinite/definite (DI_DET); and ambiguous cases (DPs that are correct/incorrect at the phrase level but not at the sentence level) (DET_ANB). As the repetition of determiners within the same phrase is a typical error, we focused our study on this error type, the R_DET. Example V.3.3 presents one automatically generated MCQ designed to deal with the correct use of determiners.

Example V.3.3 (MCQ example — Determiners)

Nire bizitzaren .... orain dela 5 urte gertatu zen.
(.... in my life took place 5 years ago.)
*a) egun zoriontsu bat (one happy day) (*key*)*

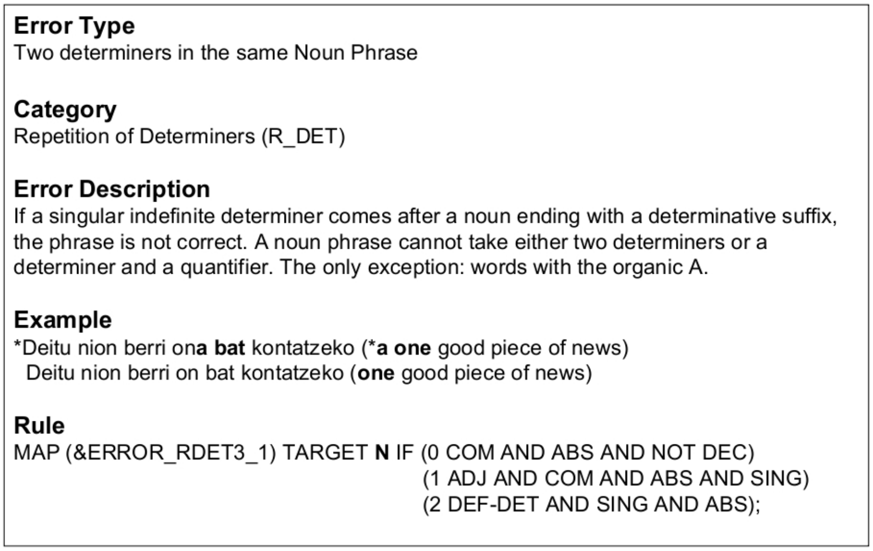
[23] For students whose mother tongue is Spanish.

*b) eguna zoriontsua bat (a one happy the day)(*distractor*)*
*c) eguna zoriontsu bat (one happy the day) (*distractor*)*
*d) egun zoriontsuak bat (the one happy day) (*distractor*)*

Section V.3.2.1 explains the methodology used to obtain the heuristics. Section V.3.2.2 presents the results of the evaluation.

V.3.2.1 Heuristics

Uria (2009) wrote rules based on CG formalism for the automatic treatment of determiner errors. Figure V.3 shows an example of a rule which reads as follows: the ERROR-tag is applied if: (i) the target noun (N) is common (COM) and does not have a declension mark; (ii) the target noun is followed by an adjective (ADJ) which is common, absolutive and singular; and (iii) the adjective is followed by a definite, absolutive and singular determiner. A total of 85 rules were written in order to detect five types of determiner error (D_DET; A_DET; R_DET; WHO_DET and DI_DET), with 58 written specifically for R_DET. Although the errors are tagged at the phrase level (i.e., the phrase containing the error is tagged), they are detected taking into account the surrounding words inside or outside the phrase and their features.



Error Type
Two determiners in the same Noun Phrase

Category
Repetition of Determiners (R_DET)

Error Description
If a singular indefinite determiner comes after a noun ending with a determinative suffix, the phrase is not correct. A noun phrase cannot take either two determiners or a determiner and a quantifier. The only exception: words with the organic A.

Example
*Deitu nion berri on**a bat** kontatzeko (***a one** good piece of news)
Deitu nion berri on bat kontatzeko (**one** good piece of news)

Rule
```
MAP (&ERROR_RDET3_1) TARGET N IF (0 COM AND ABS AND NOT DEC)
                                 (1 ADJ AND COM AND ABS AND SING)
                                 (2 DEF-DET AND SING AND ABS);
```

Figure V.3: Example rule

Erroneous phrase	Example
IZE+DET+ADJ+DET[24]	*<R_DET>Mina handia<R_DET> sentitzen nuen oinean. *I felt <R_DET>a lot of a pain<R_DET> in my foot
IZE+DET+ART	*Euskal Herria <R_DET>nazioa bat<R_DET> izan dela. *The Basque Country has been <R_DET>one a nation<R_DET>
IZE+DET+ADJ+ART	*<R_DET>Ametsa polit bat<R_DET> egin dut. *I have had <R_DET>one great a dream<R_DET>
IZE+ADJ+DET+ART	*Deitu nion <R_DET>berri ona bat<R_DET> kontatzeko. *I called him/her to tell him/her <R_DET>a good one piece of news<R_DET>
IZE+DET+ADJ+DET+ART	*<R_DET>Afaria ederra bat<R_DET> prestatu zigun. * She cooked <R_DET>a great one dinner<R_DET>
IZE+DET+ERAK	*Uste genuen <R_DET>harremana hori<R_DET> serio bihurtu zela. *We thought that <R_DET>this the relationship<R_DET> had become important
IZE+DET+ADJ+ERAK	*<R_DET>Parkea erraldoi hura<R_DET> gustatu zitzaidan. *I like <R_DET>that huge the park<R_DET>
IZE+ADJ+DET+ERAK	*<R_DET>Parke erraldoia hura<R_DET> gustatu zitzaidan. *I like <R_DET>that the huge park <R_DET>
IZE+DET+ADJ+DET+ERAK	*<R_DET>Parkea erraldoia hura<R_DET> gustatu zitzaidan. *I like <R_DET>that the huge the park <R_DET>
IZE+DET+ORO	*Eta <R_DET>gerrak guztiak<R_DET> bukatu dira. And <R_DET>all the the wars<R_DET> have ended
IZE+DET+DZG	*Orain dela <R_DET>urtea asko<R_DET> Irlandara joan nahi nuen. *<R_DET>A lot of a year<R_DET> ago I wanted to go to Ireland
DZG+IZE+DET	*<R_DET>Hainbeste oinazea<R_DET> dauka. *<R_DET>So much a pain<R_DET> has.
DZG+IZE+ADJ+DET	*<R_DET>Zenbait istorio interesgarria<R_DET> kontatu dizkigu. *<R_DET>Some an interesting stories<R_DET> has told us.
DZG+IZE+DET+ADJ+DET	*<R_DET>Zenbait istorioa interesgarria<R_DET> kontatu dizkigu. *<R_DET>Some an interesting story<R_DET> has told us.
DZG—NOLGAL—NOLARR+IZE+DET	*Vignemal <R_DET>edozein mendia<R_DET> baino politagoa da niretzat. *Vignemal is more beautiful than <R_DET>any one mountain<R_DET> to me
DZG—NOLGAL—NOLARR+IZE+DET+ADJ+DET	*<R_DET>Zein plana polita<R_DET> egin genuen. *<R_DET>What a nice a plan<R_DET> we made.
ZBKI+IZE+DET	*<R_DET>Bi liburua<R_DET> irakurri ditut. *I have read <R_DET>two a book<R_DET>
ZBKI+IZE+DET+ADJ	*<R_DET>4 herria desberdin<R_DET> pasatu genuen. *We crossed <R_DET>four different a town<R_DET>
ZBKI+IZE+ADJ+DET	*<R_DET>4 herri desberdina<R_DET> pasatu genuen. *We crossed <R_DET>four a different town<R_DET>
ZBKI+IZE+DET+ADJ+DET	*<R_DET>4 herria desberdina<R_DET> pasatu genuen. *We crossed <R_DET>four a different a town<R_DET>

Table V.20: Erroneous determiner phrases

The rules created for the automatic detection of determiner errors were the basis for generating distractors and exercises relating to the correct and incorrect use of determiners (the topic of the items). In this approach, we focused on the repetition of the determiner (R_DET), as it is one of the most common types of error and because it is not difficult to detect automatically, because in Basque it is not possible for more than one determiner to appear

[24] IZE: noun; DET: determiner; ADJ: adjective; ART: article; ERAK: demonstrative; DZG: indefinite article; ZBKI: number

within the same phrase.

Thus, the 58 rules defined by Uria (2009) were the basis for defining the heuristics. For this purpose, the rules were grouped according to the erroneous phrase they detected. Different heuristics were defined for each sub-group. Table V.20 summarises the erroneous phrases addressed, plus an example of the error.

The main idea was to take advantage of the detected errors and to establish heuristics based on them. The basis for defining the heuristics of each particular sub-group was the error types of the rest of the sub-groups and the rest of the determiner error types. In addition, some other error types which are closely related to the problems that students experience with phenomena connected with determiner were also considered (e.g., the incorrect use of declension cases). We integrated this type of error in order to encourage advanced learners by increasing the difficulty of the items so that they had to choose between the different candidates in more detail. However, the distractors with heuristics based on some kind of determiner error were given preference by means of a higher weight value.

In addition to generating distractors, as a determiner error is the starting point, this type of item also needs a correct answer. The correct answer is automatically generated based on the rule information. For each error type, Table V.21 shows its corresponding correct answer as well as the candidate distractors with the most weight. Appendix D presents all of the defined distractors for each error type.

For instance, when two determiners are detected in the same NP (*mendia bat (*one a mountain)), the system detects that there is an R_DET error (second row of Table V.21). Within the detected phrase, the noun also contains a determiner, so the corresponding correct phrase is mendi bat (one mountain); a noun heads the phrase, followed by the determiner (the article).[25] In addition to correcting the error in order to generate the corresponding correct answer, the system is able to generate (among other things) a distractor which changes the number of the noun (D1), another one which changes the article of the phrase (D2) and another one which replaces the declension case of the noun with the declension case of the article (D3) if it is not the absolutive case. As previously mentioned, the complete list of the candidate distractors can be found in Appendix D.

[25] From the point of view of generative linguistics, the determiner, in general, appears in the last position in the NP.

Error type			
IZE+DET+ADJ+DET	**C**[25]:IZE+ADJ+DET **D3**:IZE+DET+[ADJ+DET]*[26] **D6**:[IZE+DET]r[27]+ADJ+ABS	**D1**:IZE+ADJ **D4**:[IZE+DET]*+ADJ+DET **D7**:[IZE+DET]r[28]+ADJ+DET	**D2**:IZE+DET+ADJ **D5**:[IZE+DET]*+[ADJ+DET]*
IZE+DET+ART	**C**:IZE+ART **D3**:[IZE+DET]r+ART	**D1**:[IZE+DET]*+ART	**D2**:IZE+DET+[ART]f[29]
IZE+DET+ADJ+ART	**C**:IZE+ADJ+ART **D3**:IZE+DET+[ADJ]*+ART **D6**:IZE+DET+ADJ+[ART]f	**D1**:IZE+DET+ADJ+DET+ART **D4**:[IZE+DET]*+ADJ+ART **D7**:[IZE+DET]r+ADJ+ABS+ART	**D2**:IZE+ADJ+DET+ART **D5**:[IZE+DET]*[ADJ]*+ART **D8**:[IZE+DET]r+ADJ+ART
IZE+ADJ+DET+ART	**C**:IZE+ADJ+ART **D3**:IZE+[ADJ+DET]*+ART **D6**:IZE+ADJ+DET+[ART]f	**D1**:IZE+DET+ADJ+DET+ART **D4**:[IZE]*+ADJ+DET+ART **D7**:[IZE]r+ADJ+ABS+ART	**D2**:IZE+DET+ADJ+ART **D5**:[IZE]*+[ADJ+DET]*+ART **D8**:[IZE]r+ADJ+DET+ART
IZE+DET+ADJ+DET +ART	**C**:IZE+ADJ+ART **D3**:IZE+DET+[ADJ+DET]*+ART **D6**:IZE+DET+ADJ+DET+[ART]f	**D1**:IZE+ADJ+DET+ART **D4**:[IZE+DET]*+ADJ+DET+ART **D7**:[IZE+DET]r+ADJ+ABS+ART	**D2**:IZE+DET+ADJ+ART **D5**:[IZE+DET]*+[ADJ+DET]*+ART **D8**:[IZE+DET]r+ADJ+DET+ART
IZE+DET+ERAK	**C**:IZE+ERAK	**D1**:[IZE+DET]*+ERAK	**D2**:[IZE+DET]r+ERAK
IZE+DET+ADJ+ERAK	**C**:IZE+ADJ+ERAK **D3**:IZE+DET+[ADJ]*+ERAK **D6**:[IZE+DET]r+ADJ+ABS+ERAK	**D1**:IZE+DET+ADJ+DET+ERAK **D4**:[IZE+DET]*+ADJ+ERAK **D7**:[IZE+DET]r+ADJ+ERAK	**D2**:IZE+ADJ+DET+ERAK **D5**:[IZE+DET]*+[ADJ]*+ERAK
IZE+ADJ+DET+ERAK	**C**:IZE+ADJ+ERAK **D3**:IZE+[ADJ+DET]*+ERAK **D6**:[IZE]r+ADJ+ABS+ERAK	**D1**:IZE+DET+ADJ+DET+ERAK **D4**:[IZE]*+ADJ+DET+ERAK **D7**:[IZE]r+ADJ+DET+ERAK	**D2**:IZE+DET+ADJ+ERAK **D5**:[IZE]*+[ADJ+DET]*+ERAK
IZE+DET+ADJ+DET +ERAK	**C**:IZE+ADJ+ERAK **D3**:IZE+DET+[ADJ+DET]*+ERAK **D6**:[IZE+DET]r+ADJ+ABS+ERAK	**D1**:IZE+ADJ+DET+ERAK **D4**:[IZE+DET]*+ADJ+DET+ERAK **D7**:[IZE+DET]r+ADJ+DET+ERAK	**D2**:IZE+DET+ADJ+ERAK **D5**:[IZE+DET]*+[ADJ+DET]*+ERAK
IZE+DET+ORO	**C**:IZE+ORO **D3**:[IZE+DET]r+ORO	**D1**:[IZE+DET]*+ORO	**D2**:IZE+DET+[ORO]f
IZE+DET+DZG	**C**:IZE+DZG	**D1**:[IZE+DET]*+DZG	**D2**:[IZE+DET]r+DZG
DZG+IZE+DET	**C**:DZG+IZE	**D1**:DZG+[IZE+DET]*	**D2**:DZG+[IZE+DET]r
DZG+IZE+ADJ+DET	**C**:DZG+IZE+ADJ **D3**:DZG+IZE+[ADJ+DET]* **D6**:DZG+[IZE]r+ADJ+ABS	**D1**:DZG+IZE+DET+ADJ+DET **D4**:DZG+[IZE]*+ADJ+DET **D7**:DZG+[IZE]r+ADJ+DET	**D2**:DZG+IZE+DET+ADJ **D5**:DZG+[IZE]*+[ADJ+DET]*
DZG+IZE+DET+ADJ +DET	**C**:DZG+IZE+ADJ **D3**:DZG+IZE+DET+[ADJ+DET]* **D6**:DZG+[IZE+DET]r+ADJ+ABS	**D1**:DZG+IZE+ADJ+DET **D4**:DZG+[IZE+DET]*+ADJ+DET **D7**:DZG+[IZE+DET]*+ADJ+DET	**D2**:DZG+IZE+DET+ADJ **D5**:DZG+[IZE+DET]*+[ADJ+DET]*
DZG|NOLGAL|NOLARR +IZE+DET	**C**:DZG|NOLGAL|NOLARR +IZE	**D1**:DZG|NOLGAL|NOLARR +[IZE+DET]*	**D2**:DZG|NOLGAL|NOLARR +[IZE+DET]r
DZG|NOLGAL|NOLARR +IZE+DET+ADJ+DET	**C**:DZG|NOLGAL|NOLARR +IZE+ADJ+DET **D3**:DZG|NOLGAL|NOLARR +IZE+DET+[ADJ+DET]* **D6**:DZG|NOLGAL|NOLARR +[IZE+DET]r+ADJ+ABS	**D1**:DZG|NOLGAL|NOLARR +IZE+ADJ **D4**:DZG|NOLGAL|NOLARR +[IZE+DET]*+ADJ+DET **D7**:DZG|NOLGAL|NOLARR +[IZE+DET]r+ADJ+DET	**D2**:DZG|NOLGAL|NOLARR +IZE+DET+ADJ **D5**:DZG|NOLGAL|NOLARR +[IZE+DET]*+[ADJ+DET]*
ZBKI+IZE+DET	**C**:ZBKI+IZE	**D1**:ZBKI+[IZE+DET]*	**D2**:ZBKI+[IZE+DET]r
ZBKI+IZE+DET+ADJ	**C**:ZBKI+IZE+ADJ **D3**:ZBKI+IZE+DET+[ADJ]* **D6**:ZBKI+[IZE+DET]r+ADJ+ABS	**D1**:ZBKI+IZE+DET+ADJ+DET **D4**:ZBKI+[IZE+DET]*+ADJ **D7**:ZBKI+[IZE+DET]r+ADJ	**D2**:ZBKI+IZE+ADJ+DET **D5**:ZBKI+[IZE+DET]*+[ADJ]*
ZBKI+IZE+ADJ+DET	**C**:ZBKI+IZE+ADJ **D3**:ZBKI+IZE+[ADJ+DET]* **D6**:ZBKI+[IZE]r+ADJ+ABS	**D1**:ZBKI+IZE+DET+ADJ+DET **D4**:ZBKI+[IZE]*+ADJ+DET **D7**:ZBKI+[IZE]r+ADJ+DET	**D2**:ZBKI+IZE+DET+ADJ **D5**:ZBKI+[IZE]*+[ADJ+DET]*
ZBKI+IZE+DET+ADJ+DET	**C**:ZBKI+IZE+ADJ **D3**:ZBKI+IZE+DET+[ADJ+DET]* **D6**:ZBKI+[IZE+DET]r+ADJ+ABS	**D1**:ZBKI+IZE+ADJ+DET **D4**:ZBKI+[IZE+DET]*+ADJ+DET **D7**:ZBKI+[IZE+DET]r+ADJ+DET	**D2**:ZBKI+IZE+DET+ADJ **D5**:ZBKI+[IZE+DET]*+[ADJ+DET]*

Table V.21: Generation of items regarding determiners

Even if each phrase has its own heuristics, they can be grouped into more general categories. This is, in fact, what the functions of the question model express. Table V.22 summarises the functions defined in the question model.

[26] **C**: Correct answer; **D1**: Candidate Distractor 1; **D2**: Candidate Distractor 2; **D3**: Candidate Distractor 3; **D4**: Candidate Distractor 4; **D5**: Candidate Distractor 5; **D6**: Candidate Distractor 6; **D7**: Candidate Distractor 7.

[27] *: Change the number.

[28] r: Replace the declension suffix with the one from the other component of the phrase, if it is not absolutive case.

[29] f: Change the article.

Function
change_number(language,from,to)
replacement(language,from,to)
change_article(language,from,to)

Table V.22: The representation of heuristics in the question model for determiners

Example V.3.4 shows an instance of the change_article function in the question model. This example expresses that, given the input word bat (one) that is, a definite article, by means of the function change_article, the system generates the corresponding indefinite article asko (a lot). Although the example given shows the minimum information needed by the system in the generation step, the complete candidate distractor should be mendia asko (a lot a mountain) (D2 in the second row of Table V.21).

Example V.3.4

```
<heuristic>
  <type>determiner_error</type>
  <function>change_article(dzh,dzg)</function>
  <input> bat </input>
  <output> asko </output>
</heuristic>
```

The rule information is used by the system to create the correct answer. Following on from the previous example, ArikIturri gives the Basque morphological generator the information needed in order to obtain the correct answer mendi bat (one mountain). Once the answer is generated, it is represented by means of the <answer> tag in the question model. In addition, this type of item includes the <artificial> tag, which expresses that the correct answer has been artificially generated (see chapter III). Example V.3.5 shows the parts of the instance that represent the aforementioned information.

Example V.3.5

```
<answer>
  <word pos="0"> mendi </word>
  <word pos="1"> bat </word>
  <topic_info>
    ...
    <artificial>true</artificial>
    ...
  </topic_info>
  ...
</answer>
```

Although the items presented in this section are based on the previously detected determiner errors, there is also the option of generating items from a correct key. This time, there is no need to generate the correct answer, but only the candidate distractors. For this reason, the heuristics are based on different determiner errors, as well as on declension errors. Basically, the same criteria were used when defining the heuristics. The evaluation presented below uses as its starting point determiner errors. All of the relevant information can be found in Appendix D.

V.3.2.2 Evaluation

In a completely real scenario, each learner would have to answer the test based on his or her own text and errors. This evaluation, however, was carried out with a previously written text. One text written by a low-level learner who was asked to write a description of the happiest day in her/his life was the source text for this test. The low level is the level at which the most R_DET errors have been detected (proportionally and with a statistically significant difference) in the learner corpus (Uria, 2009). In addition, the number of texts within the Basque language learner corpus with a theme which is on this level is fairly high (258). These two features gave us a way of comparing our results with the analysis done of the learner corpus (Uria, 2009).

Although the texts comprising this sub-group of the corpus contain errors, none had enough R_DET errors to be tested. Therefore, the errors were created manually for the present evaluation. This creation was not carried out at random. The linguist who wrote the rules was responsible for simulating the different types of R_DET errors which were previously observed and detected in the corpus as a whole. The expert had to establish different examples of R_DET errors which could be detected by different rules,

Item	D1	C	D2	D3
1	R_DET	egun zoriontsu bat (one happy day)	R_DET	R_DET
2	R_DET	bost urte (five years)	W_DET	W_DET
3	R_DET	rock zale bat (one rock fan)	R_DET	R_DET
4	R_DET	urte batean (in one year)	WO_DET +W_DET	R_DET
5	R_DET	egunkari batean (in a newspaper)	DEK	R_DET
6	R_DET	Jende askok (lot of people)	R_DET	R_DET
7	R_DET	berri hori (this piece of news)	R_DET	DEK
8	R_DET	zelai bat (one field)	R_DET	W_DET
9	R_DET	lagun batzuekin (with some friends)	R_DET	DEK
10	R_DET	garagardo batzuk (some beers)	R_DET	WO_DET +W_DET
11	R_DET	piano baten (in a piano)	R_DET	DEK
12	R_DET	urte asko /lot of years)	DEK	R_DET
13	R_DET	piano bat (one piano)	WO_DET +W_DET	R_DET
14	R_DET	klase batzuk (some lessons)	R_DET	R_DET
15	R_DET	egun batean (one day)	R_DET	DEK

Table V.23: Features of each item

obtaining the highest level of casuistry possible in order to apply heuristics which were as different as possible. In total, the linguist marked 15 NPs as erroneous out of a total of 68 within text, that is to say, 22.05% of the NPs.

Once the NPs containing the determiner error types had been created, ArikIturri generated for each of them the corresponding correct answer and two more distractors. As previously presented in Table V.21, all of the error types resulted in the generation of more than two candidate distractors. In this evaluation, although the ones with the highest weighting were given preference, a random feature was added in order to avoid the repetition of the same type of candidate distractor all the time. Each row of Table V.23 shows the type of error which was generated automatically for each candidate distractor (columns 4 and 5) together with the corresponding correct answer (column 3) for each manually generated R_DET error.

Therefore, as shown in Table V.23, not all of the candidate distractors generated by ArikIturri were R_DET errors. In some cases, the system generated WO_DET and W_DET determiner errors. In other cases, it applied a combination of both error types in order to generate the candidates. Fi-

#Correct	**#Students**	**Percentage (%)**
15	4	13.33
14	6	20.00
13	5	16.67
12	3	10.00
11	1	3.33
10	2	6.67
9	2	6.67
8	1	3.33
7	1	3.33
6	1	3.33
5	1	3.33
4	2	6.67
1	1	3.33

Table V.24: Number of students per acceptance rate

nally, errors relating to declension cases were also considered when generating the distractors. Nonetheless, the most commonly generated error type was R_DET.

With regard to the collected learner corpus, the percentage of R_DET errors in the low-level sample is 0.77%. If we go into greater detail and look at the sample of texts in which students describe the happiest day in their lives, the percentage that corresponds to R_DET errors is 1.09%.

The generated test was given to Basque language learners whose mother tongue was Spanish. Thirty low-language-level learners took part in the experiment. Table V.24 groups the students by the number of items they answered correctly. Based on the fact that 60% of the students (18) made three errors or fewer, it is clear that the generated test was not very difficult for a high percentage of the students.

Although not all of the students made mistakes, the number of errors made by some of them increased when the learners were forced to complete such a test. There was a significant difference in percentages when comparing the number of errors made by these students with the errors collected in the learner corpus.

Based on the results of the students who made at least four mistakes, the percentage of R_DET errors per NP was 5.88% or higher. In contrast, as previously presented, the percentage in learner corpus was 1.09%. In addition, there are no examples of documents with a significantly higher number of errors in comparison to others.

These results and our experience in the field led us to propose several hypotheses:

- Offering this type of test leads to the emergence of errors that students do not produce in a writing task;

- Learners tend to write in a simpler way in order to avoid making errors.

These hypotheses suggest some good reasons to continue investigating this research line.

V.4 Distractor generation: automatically extracted patterns

We have already mentioned the fact that ArikIturri makes use of different heuristics in order to create distractors for questions. In the previous section, the information used to define the heuristics was created manually, but it could also be generated automatically. The experiments presented below explore this new area of research.

In this Basque language learning scenario, the aim was to create items to test the correct use of Basque declension cases, as displayed in example V.4.1.

Example V.4.1 (Error correction example)

Hainbat ariketaren bidez gure gorputzaren ***blokeoarekin*** *askatu dugu.*

((we) have released ***with*** *the stiffening of our bodies by means of some exercises.)*

V.4.1 Automatic extraction of patterns to define heuristics

The heuristics employed in sections V.3.1 and V.3.2 were defined based on expert knowledge. This new approach, in contrast, aims to define an automatic process not only for the generation of distractors, but also for the generation of heuristics. That is to say, the study presented in this section explores the option of defining heuristics based on automatically extracted grammatical information. As in the previous sections, ArikIturri employs grammatical information in order to generate items relating to Basque grammar, and more specifically, items dealing with the declension of nouns. For this reason, we first extracted some patterns that are used as the basis for some rules. These rules form the basis of the generation of distractors, and they represent some of the possible unsuitable combinations from a linguistic point of view.

The basis of the automatic extraction of patterns to define heuristics comes from the work of Aldezabal *et al.* (2003), in which a finite-state syntactic grammar was developed in order to join verb instances and their corresponding syntactic dependents (arguments and adjuncts) from journalistic corpora. This syntactic grammar had a score of 87% of precision and 66% for recall, and their system obtained 688 different patterns for 640 verbs. For each verb, more than one of the 688 different patterns can occur.

The patterns which represent the knowledge extracted from automatically analysed corpora were obtained at the simple sentence level. Moreover, Aldezabal *et al.* (2003) automatically retrieved elided cases in order to reflect them in the patterns. Each of the patterns offers the following information: (i) the syntactic dependents; (ii) the auxiliary type; and (iii) the number of instances. For instance, one of the 143 extracted patterns relating to the verb askatu (to release) is:

48 askatu: DU: ABS + ERG + INE

Based on journalistic corpora, the system developed by Aldezabal *et al.* (2003) found 48 matches for the DU: ABS + ERG + INE pattern for the verb askatu (to release). This reflects the number of times that the absolutive, the ergative and the inessive occur with the auxiliary DU.

This type of patterns was the basis for the automatic generation of the heuristics. In this way, when the *distractor generator* generated a distractor, the patterns were automatically extracted and the distractor pattern that had been created could be compared. If a matching was detected, the distractor could not be considered a candidate distractor and the question was automatically rejected.

The experiments previously carried out in section V.3.1.2 offer us the chance to compare manually generated heuristic with automatic patterns. Moreover, as the questions have already been manually evaluated, we can study the measure of success of the patterns. In any case, the proposal presented here is applicable to any declension case.

Once the clauses of the questions have been extracted, it is important to specify which phrases are going to be taken into consideration when matching them to the patterns. Thus, we studied different criteria for comparing the automatically extracted patterns with the phrases of the questions generated by ArikIturri.

Example V.4.1 presents an error correction item for a topic relating to the correct use of the absolutive. The phrase containing the correct answer gure gorputzaren blokeoa (the stiffening of our bodies) is absolutive and has been transformed into the sociative (SOZ) case in order to generate the distractor gure gorputzaren blokeoarekin. The phrase Hainbat ariketaren bidez refers to the instrumental case (INS) and the auxiliary for the verb askatu is DU.

The auxiliary DU for the verb askatu tells us there is a subject (ERG) as well as a direct object (ABS).

The criteria used to compare the clause of the question with the patterns can be summarised as follows:

- *Criterion 1*: Compare the patterns with the declension cases/phrases that appear explicitly in the clause. In the previous example, in the case of the distractor, DU: INS + SOZ would be compared with the patterns from Aldezabal *et al.* (2003). As there is no match, ArikIturri would create a distractor.

If we want to take into account the phrases containing some of the given declension cases that occur in a clause plus those which are elided, we can follow two different options:

- *Criterion 2*: Contrast the cases which have been elided, if they are not part of the topic. In the example, the system would compare DU: INS + SOZ + ERG with the automatic patterns. That is to say, we would take into consideration the ergative case because it is elided and as it is not the topic. In contrast, we would not consider the absolutive case because it is the topic of the question. In this case, the system would generate a distractor, as the distractor pattern does not match any of the 143 patterns extracted for the verb askatu;

- *Criterion 3*: Include all of the elided cases. In the example, we would compare the paradigm DU: INS + SOZ + ERG + ABS with the patterns. As this distractor pattern exists in Aldezabal *et al.* (2003), the system would not generate a distractor.

In the next two sections, we explain the two strategies which can be followed in the automatic generation of heuristics. The first one has been developed in order to generate complex sentence questions, while the second one is carried out in order to create simple sentence questions.

V.4.2 Heuristics based on patterns used to generate complex sentence questions

This first attempt compared the questions which were evaluated in section V.3.1.2 with the patterns which were extracted automatically from journalistic corpora. For this purpose, these steps were followed:

1. Obtain a sample of the error correction questions relating to the sociative, inessive, ergative, dative or absolutive cases. This was the same sample as the one used in section V.3.1.2 (a sample of 25% of the error correction questions of the first experiment);

2. Extract the simple sentence in which the topic appears from each question. This task was performed manually. When the topic was part of the subordinate clause, the subordinate clause was manually transformed into a main clause;

3. Compare the questions (at the clause level) with the patterns in order to observe the acceptance rate if we generated heuristics automatically based on the automatic patterns.

As there was a high rate of agreement between the editors in the previous experiments (cf., section V.3.1.2), we first performed a study of the heuristics used in the generation of well-formed questions. We compared the questions which were accepted in the first experiment from section V.3.1.2 with the patterns, i.e., the information in the well-formed questions was divided in order to compare both the correct answer and the distractors.

If we applied *criterion 1*, we might expect low results, as it only takes into account the explicit phrases within the question, while the patterns automatically assign the explicit declension cases of the verb as well as those that are elliptic. Nevertheless, low results were obtained in the case of the correct answers, but not in the case of the distractors.

Table V.25 shows the results relating to the three different criteria when comparing the data from the questions accepted in section V.3.1.2.

	Criterion 1 (%)	Criterion 2 (%)	Criterion 3 (%)
Correct answer in the 1st experiment	66.27	93.59	93.59%
Distractor in the 1st experiment	69.94	66.46	37.89%

Table V.25: Accepted questions

For instance, if we had, as the basis, automatically extracted knowledge (patterns) when using *criterion 1*, 66.27 out of the 100 correct answers accepted by the editor in section V.3.1.2 would also be considered as correct answers. In addition, 69.94% refer to the clauses of the questions that were accepted by the editor as distractors. Almost 70% of the distractors of the questions would also be considered as distractors if the patterns were used for the automatic generation of the heuristics.

Regarding the results obtained for *criterion 2*, we can conclude that they are more realistic and better. As the correct answers are extracted from the source sentence of the corpus, they are presumably correct.

The number of questions which were created by ArikIturri and rejected by the editors is not troubling, as it is a low percentage.[30] However, the patterns can also be compared with these rejected questions in order to observe whether the distractors which were rejected by the human editors would not be created if the patterns formed the basis of the heuristics.

Table V.26 represents the percentages of rejected distractors in the first experiment of section V.3.1.2 for the three different criteria.

As in the case of accepted questions, in this case, the method of comparing the questions with the patterns is threefold. This time, the given percentages have a different meaning. As the questions were rejected by the editor, it

[30] This was 17.29% in the first experiment for the error correction question types, and 6.78% in the second experiment of section V.3.1.2.

	Criterion 1 (%)	Criterion 2 (%)	Criterion 3 (%)
NOT distractor in the 1st experiment	15.38	19.23	48.00

Table V.26: Rejected questions

is assumed that the rejected distractors were not proper.[31] If we used the automatically extracted patterns to create heuristics and compared them only with the explicit phrases of the clause, 15.38% of the distractors would be considered improper distractors in the case of *criterion 1*. In this case, better results are obtained from the third comparison.

We noted some aspects that could affect the results: the error rate of the patterns; the corpus; and the working unit. The error rate of the patterns may alter the results, as their precision level is 87%. Therefore, 13% of the patterns obtained by Aldezabal *et al.* (2003) are incorrect. The corpus should also be considered, because different corpora have been used in both works. In the case of ArikIturri, the corpus is focused on language learning, while for the automatic extraction of patterns, the corpus is composed of newspaper texts. Finally, the experiments described in section V.4.2 were carried out using complex sentences while the extracted patterns refer to simple ones. Therefore, the working unit could have influenced the results.

V.4.3 Heuristics based on patterns used to generate simple sentence questions

The fact that the working unit could affect the results led us to carry out a new experiment in order to obtain heuristics based on patterns designed to generate simple sentence questions. This time, we presented the editors with new questions to be evaluated. These questions were simple sentences which had been manually extracted from the complex sentences used in the previous experiments. As we have noted, we used a sample of 25% of the error correction questions which were related to the sociative, inessive, ergative, dative and absolutive cases.

The editors evaluated the questions and accepted 75.21% of them. This is smaller compared to the acceptance rate in the first (82.71%) and sec-

[31] The error correction question type has just one distractor.

ond (93.22%) experiments. These results correspond to the generated error correction questions relating to declension cases. The only difference lies in the evaluated sentences: the ones from the first and second experiments were complex sentences, while in this final experiment, they were simple sentences.

Once we obtained a set of manually evaluated questions, we compared them with the automatically generated patterns. This time, we used the three aforementioned criteria. Table V.27 shows the pattern accuracy, taking into account the accepted questions (75.21%).

	Criterion 1 (%)	Criterion 2 (%)	Criterion 3 (%)
Correct in the 1st experiment	100	100	100
Distractor in the 1st experiment	79.19	78.05	77.64

Table V.27: Accepted questions at the simple sentence level

The results from this experiment are better than those shown in Table V.25. In all cases, the patterns would consider all of the correct answers. Regarding the distractors, the best results were obtained from the first comparison, although there are no significant differences between the three evaluations. Moreover, the results are closer to the error rate of the automatically extracted patterns than which is shown in Table V.25.

In the case of the rejected questions, the same equivalence was carried out. Table V.28 displays the results.

	Criterion 1 (%)	Criterion 2 (%)	Criterion 3 (%)
NOT distractor in the 1st experiment	38.46	38.46	40.00

Table V.28: Rejected questions at the simple sentence level

Although the results are better than those shown in Table V.26, they are still fairly poor. In the case of *criterion 1* and *criterion 2*, the results are twice as good, but still poor.

The comparison of the results obtained in the evaluations shows us that a clause which can be considered a question at the complex sentence level does not always translate into a question at the simple sentence level, and vice versa. Moreover, the editors took into account the elided sentence elements when evaluating the questions at the simple sentence level. Finally, we can also conclude that the best results were obtained from *criterion 2*.

Therefore, as it is not always possible to define heuristics with the help of experts, this method gives us a way of doing so.

In conclusion

In this chapter, we have presented the ways in which ArikIturri employs grammatical information in different steps of the generation of items. We have paid special attention to the stem and distractor generation tasks. For this purpose, we first studied the corpora and NLP tools within reach. Based on this analysis, we designed four experiments to investigate the applicability of integrating the use of grammatical information into the creation of items.

CHAPTER VI

Using semantics when generating test items

This chapter presents various experiments in which ArikIturri applies semantic information in order to generate items. More specifically, the semantic information is used within the distractor generation task. In addition, the multilinguality of the system is proven when generating English tests. Finally, experiments which are focused on the science domain simulate the testing process as a whole.

VI.1 Introduction

This chapter presents experiments relating to the use of semantic information. In this approach, ArikIturri exploits such information within the *distractor generation* module in order to generate distractors which are semantically similar to the key for MCQs. We have designed two scenarios in order to test students' knowledge of vocabulary. The experiment presented in section VI.3 aims to prove the multilinguality of the system, and the experiment presented in section VI.4 aims to simulate the entire testing process of items in a real scenario.

In the first scenario, explained in section VI.3, the experts had to evaluate MCQs designed to test vocabulary, like the one presented in example VI.1.1. We restricted the test to English verbs which appear in the AWL and the generated items were presented in isolation. These constraints were previously established due to the option of evaluating the generated items with English teachers who actually make use of this type of test in their classes. Thus, evaluating the items with experts who were familiar with the AWL made the evaluation more real due to their experience.

Example VI.1.1 (MCQ to test the verb "respond" from the AWL)
Certainly, people.... to whether they perceive the world as threatening or reassuring.

1 a. respond b. identify c. function d. interpret

In the second scenario, explained in section VI.4, the MCQs are embedded in the text, as example VI.1.2 presents. In this scenario, we predefined three constraints. First, the aim was to work with the Basque language. Second, we set a domain-specific scenario: science and technology. Finally, we looked at the profile of students from the 21^{st} century. As regards these domain-specific tests, we had the option of receiving help from experts and testing the vocabulary relating to science and technology with OSE students (second grade). As a consequence, the generated items were tested as part of a entire text, as this is the type of test that teachers conduct in class.

Example VI.1.2 (MCQs embedded in the source text — Basque)
Espazioan itzalkin erraldoi bat ezartzeak, bestalde, Lurrari...1... egingo lioke, poluitu gabe. Siliziozko milioika disko ...2... bidaltzea da ikertzaileen ideia. Paketetan jaurtiko lirateke, eta, behin diskoak zabalduta, itzalkin-itxurako egitura handi bat osatuko lukete. Hori bai,...3... handiegiak izango lituzke.[1]

1	*a. babes*	*b. aterki*	*c. defentsa*	*d. itzala*
2	*a. unibertsora*	*b. izarrera*	*c. galaxiara*	*d. espaziora*
3	*a. kostu*	*b. prezio*	*c. eragozpen*	*d. zailtasun*

Both scenarios were defined in order to test vocabulary, and so both are based on the idea of **semantic similarity** or **relatedness**. Even if these terms are sometimes used interchangeably, they are not identical terms.

> Semantic similarity represents a special case of semantic relatedness: for example, the terms *cars* and *gasoline* would seem to be

[1] The establishment of a huge parasol will be like a...1... to the Earth, without pollution. Researchers intend to sent millions of silicon disks...2... They would be thrown in packets, and once the discs expand, they would constitute a huge parasol-shaped structure. However, the...3... would be too big.

1	a. protection	b. umbrella	c. defense	d. shadow
2	a. to the universe	b. to the star	c. to the galaxy	d. to the space
3	a. cost	b. price	c. difficulty	d. hardness

> more closely related than, say, *cars* and *bicycles*, but the latter pair are certainly more similar (Resnik, 1995).

As it is expounded by Zesch and Gurevych (2010), the terms are associated by means of "classical" taxonomy relations like hyponymy (science/ natural science) or meronymy (finger/hand). In addition, terms can also be connected through "nonclassical" taxonomy relations (Morris and Hirst, 2004), as with *cars* and *gasoline*. Therefore, semantic relatedness indicates the degree of association via any type of semantic relationships, while semantic similarity takes into account only classical relations in order to determine the degree of similarity between two terms.

Measures of relatedness or similarity are used in many NLP tasks such as information extraction, word sense disambiguation, textual entailment and error correction. The methods which are used to solve this problem can be divided into two main categories: **knowledge-based** approaches and **corpus-based** approaches. The former makes use of knowledge resources to measure similarities such as dictionaries, thesauri, WordNets or Wikipedia and these resources can be classified based on their creators: linguists and "crowds" (Zesch and Gurevych, 2010). The corpus-based approach takes into account the distributional properties of words from corpora, and that is why this type of measure is also known as lexical distributional similarity measures.

Thus, in our case, the first decision to make before implementing any heuristic to deal with semantics was to set the most convenient technique for our research line. Looking at the few studies which involve experiments focused on semantics and the automatic generation of distractors, we found that, for instance, Pino *et al.* (2008) employ WordNet to measure semantic similarity and that Smith *et al.* (2009) used distributional information from a corpus. In addition, there are some studies which exploit both approaches, such as Mitkov *et al.* (2009). Although all of these studies present interesting approaches, there is no one which outperforms the rest. In fact, it is not possible to compare the different proposals, as the sources used are completely different.

Based on the availability of the resources, the distributional similarity measure was set as the starting point for the distractor generation task in our scenarios. As will be explained in the following sections, English verb candidate distractors were obtained based on the **information radius** measure, while the Basque scientific candidate distractors were extracted from a **latent semantic analysis (LSA)** model. In addition, a **graph-based** approach was also studied in the science domain. Before going into detail about the scenarios, section VI.2 presents the two distributional similarity methods and the graph-based approach.

VI.2 Semantic relatedness methods

VI.2.1 Distributional similarity

We start from the premise of lexical distributional similarity, according to which two terms are said to be similar if they appear in similar contexts. The context can be modelled at different levels, meaning that co-occurrence can be defined with regard to documents, n-grams, bags of words or even the words upon which the target term is somewhat grammatically dependent.

These similarity measures can be conceived as measures of vector similarity (Manning and Schütze, 1999), and the two terms to be compared are represented as vectors in the corresponding multi-dimensional space. Tables VI.1, VI.2 and VI.3, taken from Manning and Schütze (1999), give examples of such multidimensional spaces.

	cosmonaut	astronaut	moon	car	truck
d_1	1	0	1	1	0
d_2	0	1	1	0	0
d_3	1	0	0	0	0
d_4	0	0	0	1	1
d_5	0	0	0	1	0
d_6	0	0	0	0	1

Table VI.1: A document-by-word matrix

The matrix represented in Table VI.1 corresponds to a document space in which entry a_{ij} contains the number of times that the word j occurs in document i. In the case of Table VI.2, the matrix represents the terms as vectors in a word space in which entry b_{ij} contains the number of times that the word j co-occurs with word i. Based on the given examples, while in terms of document space the words *cosmonaut* and *astronaut* are dissimilar because they do not appear in the same documents, in terms of word space, they are more similar because both terms co-occur with *moon*.

	cosmonaut	astronaut	moon	car	truck
cosmonaut	2	0	1	1	0
astronaut	0	1	1	0	0
moon	1	1	2	1	0
car	1	0	1	3	1
truck	0	0	0	1	2

Table VI.2: A word-by-word matrix

	cosmonaut	astronaut	moon	car	truck
Soviet	1	0	0	1	1
American	0	1	0	1	1
spacewalking	1	1	0	0	0
red	0	0	0	1	1
full	0	0	1	0	0
old	0	0	0	1	1

Table VI.3: A modifier-by-head matrix

Finally, the matrix displayed in Table VI.3 represents the nouns as vectors in terms of modifier space. That is, the nouns (or heads of NPs) are modified by the adjectives, so that entry c_{ij} contains the number of times that the head j is modified by the modifier i. For *cosmonaut* and *astronaut*, we can see that in the modifier space, they are also similar terms because they are modified by the same modifier (spacewalking). However, the term *moon* in this space is dissimilar to *cosmonaut* and *astronaut*, while in the previous spaces (the document and word spaces) they were similar. Therefore, different spaces indicate different types of similarity (Manning and Schütze, 1999).

Although different results are obtained based on the context used, there is still an open debate regarding which level of context is the "best". It depends, to a degree, on the purpose of the task in which similarity is being computed. Intuitively, as the information used in the modifier space is more fine-grained, this type of context could be beneficial in the distractor generation task. In the case of English verb tests (cf., section VI.3), we focus on the similarity between verbs based on their co-occurrence with nouns in the predicate-object relation, that is, we work with a modifier space in which the similarities between verbs are computed according to the distributional data (cf., section VI.2.1.1). More specifically, the measure is based on the co-occurrence frequencies of verbs and their distributional features (the nouns that are part of the predicate-object co-occurrence pairs). As we will see in section VI.4, the similarities obtained as regards Basque scientific terms are based on the document space (cf., section VI.2.1.2), as the available corpus was too limited to extract a reliable modifier space.

VI.2.1.1 Information radius

There are many measures based on vector space for computing similarities. The most well-known are the Dice's coefficient, the Jaccard's coefficient and the cosine coefficient, but these are not the only ones. One limitation of the Dice and Jaccard coefficients is that they work in a Boolean space, and some sensitive data could be set aside. The cosine coefficient works with real-numbered dimensions assuming a Ecludiean space, which is appropriate for normally distributed quantities, but not for counts and probabilities (Manning and Schütze, 1999). It is possible to transform the matrices presented above (see Tables VI.1, VI.2 and VI.3) into matrices of conditional probabilities in order to compute probabilistic measures. Thus, the (dis)similarity measure can be seen as the difference between two probability distributions (Manning and Schütze, 1999). Among the dissimilarity measures proposed by Dagan *et al.* (1997), we chose the information radius (or total divergence to the average, IRad) measure for the distractor generator module. In addition to the fact that the measure is symmetrical, it has no problem with infinite values and obtained the best results in comparison to Kullback-Lieber (KL) divergence and L_1 norm measures (Dagan *et al.*, 1997).

The information radius is a variant of KL divergence. While the KL divergence measure calculates how different two probability distributions are, the IRad measures measures how much information is lost if two words, verbs in our experiments, are described with their average distribution. As the vectors are obtained from a modifier space in both cases, the results depend solely on the nouns which occur with both verbs. Thus we can set:

$$p = P(n|v_1), q = P(n|v_2)$$

for which the relative entropy, $D = (p \parallel q)$, is the inverse of the distance between the distribution of p and q.

$$\mathcal{D}(p \parallel q) = D(P(n|v_1) \parallel D(P(n|v_2) = \sum_n P(n|v_1) log \frac{P(n|v_1)}{P(n|v_2)}$$

Therefore, as IRad is the average of the KL divergence of each of the two distributions to their average distribution, it can be formulated as follows:

$$IRad = D(p \parallel \frac{p+q}{2}) + D(q \parallel \frac{p+q}{2})$$

VI.2.1.2 Latent semantic analysis

In experiments in the science domain (cf., section VI.4), ArikIturri uses context words to compute similarity, deploying LSA (Landauer *et al.*, 2007). LSA has produced successful results in a number of NLP tasks such as information retrieval (Deerwester *et al.*, 1990), and in the evaluation of synonym test questions (Turney, 2001). It has also been applied in the field of education (Landauer *et al.*, 2007).

LSA is based on the concept of vector space models (VSMs) which were originally created in order to deal with text retrieval from heterogeneous texts.[2] VSMs define unique vectors for each term and document, and queries are performed by comparing the query representation to the representation of each document in the vector space (Landauer *et al.*, 2007).

In general, in order to create a VSM for LSA, a term-by-document matrix is constructed. In this, element a_{ij} of Matrix A is the frequency of the ith term in the jth document.

After that, a weighting function is applied to each nonzero element, a_{ij}, of Matrix A in order to improve retrieval performance. A common weighting function is log-entropy, which increases or decreases the importance of terms within documents and across the entire collection.

In general, the term-by-document Matrix A is considered to be sparse because it contains mainly zero entries. Thus, Matrix A is transformed into a term and document vector space through orthogonal decomposition in order to reduce its dimensions. An orthogonal matrix is one with the property $Q^TQ = I$, where Q is an orthogonal matrix, Q^T is the transpose of matrix Q and I is the identity matrix. The most popular method of obtaining them is singular value decomposition (SVD) (Landauer *et al.*, 2007) and it is represented as follows:

$$A = U \sum V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

[2] In information retrieval is also called latent semantic indexing (LSI).

where the rows of Matrix U are the type vectors and are called left singular vectors. The rows of V are the document vectors and are called right singular vectors. The nonzero diagonal elements of $\sum$ are known as the singular values and the rank of $A = r$ (Landauer *et al.*, 2007).

Given the fact that A can be written as the sum of rank 1 matrices, r can be reduced to k in order to create:

$$A_k = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Thus, Matrix A_k is the closest rank k approximation to the original matrix. SVD implicitly finds correlations between features and documents, identifying synonymous or closely related words and other relations between features in order to remove the “noise”. In conclusion, this k-dimensional vector space is the basis of the semantic structured used in LSA and so by our system.

VI.2.2 Graph-based method

The graph-based approach used to create distractors to be part of the science domain (cf., section VI.4.2) was first introduced by Agirre and Soroa (2009). The method is based on a lexical knowledge base (LKB). An LKB such as WordNet can be seen as a set of concepts and the relationships between them, plus a dictionary, which contains the list of words (typically word lemmas) which are linked to the corresponding concepts (senses). WordNet can thus be represented as a graph $G = (V, E)$ in which nodes represent concepts (v_i), and each relation between concepts v_i and v_j is represented by an edge $e_{i,j}$.

VI.2.2.1 PageRank and personalised PageRank

The PageRank random walk algorithm (Brin and Page, 1998) is a method for ranking the vertices in a graph according to their relative structural importance. The main idea of PageRank is that whenever a link from v_i to v_j exists in a graph, a vote from node i to node j is produced, and hence the rank of node j increases. In addition, the strength of the vote from i to j also depends on the rank of node i: the more important node i is, the more strength its votes will have.

Let G be a graph with N vertices $v_1, \ldots, v_N$ and d_i be the outdegree of node i; let M be an $N \times N$ transition probability matrix, where $M_{ji} = \frac{1}{d_i}$

if a link from i to j exists, and zero otherwise. Then, the calculation of the *PageRank Vector* **P** over G is equivalent to resolving Equation (VI.1).

$$\mathbf{P} = cM\mathbf{P} + (1 - c)\mathbf{v} \tag{VI.1}$$

In the equation, **v** is a $N \times 1$ stochastic vector and c is the so-called *damping factor*, a scalar value between 0 and 1. The first term of the sum of the equation models the voting scheme described at the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node, e.g., without following any of the paths on the graph.

The traditional PageRank formulation is independent of the context. The vector **v** is a stochastic normalised vector, with element values which are all equally probable in the event of random jumps. However, as pointed out by Haveliwala (2002), the vector **v** can be non-uniform and can assign stronger probabilities to certain kinds of node, effectively biasing the resulting PageRank vector towards these nodes. Agirre and Soroa (2009) call this a personalised PageRank.

In order to apply a personalised PageRank, when given an input text, e.g., a sentence, the method extracts the list of the content nouns which are in the dictionary, concentrating the initial probability mass over the context words. In this way, they can be related to LKB concepts, and as a result of the PageRank process, every LKB concept receives a score. Therefore, the resulting personalised PageRank vector can be seen as a measure of the structural relevance of LKB concepts in the presence of the input context. In our case, we used MCR 1.6 as the LKB and Basque WordNet as the dictionary.

VI.3 MCQs for English verb tests

The aim of the set of experiments expounded in this section is to generate isolated MCQs of English vocabulary, and more specifically, with English verbs from the AWL. For this reason, the candidate distractors were automatically generated by means of the information radius distributional similarity measure. However, this approach not only aims to generate the distractors automatically, but also to select the source sentences from the input corpus automatically. Section VI.3.1 presents the experiments relating to the sentence selection task. Section VI.3.2 deals with different attempts to generate distractors.

The system generates MCQs such as the one displayed in example VI.3.1.

Example VI.3.1 (Example of an MCQ for English verbs)
1.- Hare's account of the events......, and many obscure points in the history of the discovery have subsequently been investigated.

a. has been published	*c. has been issued*
b. has been commissioned	*d. has been contributed*

It is also important to remark that, as the system was first developed for Basque, its adaptation to English is an illustration of its multilingual portability. The generation of items was made possible by parsing the input texts using Connexor Machinese Syntax (Tapanainen and Jarvinen, 1997). The output of this analysis was a syntax tree, in which chunks are not represented explicitly. To this end, we implemented a post-processing module for the English version which takes the output of the parser and produces the chunks needed for the generation of test items.[3] Unlike the original system for Basque, which is an agglutinative language, no morphological generator is needed for English. As a consequence, a database for storing all of the possible word forms has been included in the system.

The sentence selection and distractor generation tasks explained in the subsequent sections follow the same process with a view to arriving at new heuristics. We first established a baseline system which was manually evaluated by an expert of English as a Foreign Language (EFL) teacher; her opinion served as the gold-standard. Then, an experiment was defined in order to improve the baseline system with the new heuristic rules. Finally, the results of the experiment are compared with the gold-standard.

The basis for all of the experiments presented below was 200 MCQs chosen at random from a larger sample. The generated items came from two different corpora: the BNC, a general corpus, and a mystery novel of 140,000 words that can be seen as a specific corpus. The reason for evaluating the methods in different domains was motivated by the idea that the domain could influence the acceptance rate of the items.

[3] Although it is possible not to represent the chunk information explicitly (cf., chapter III), we consider it interesting to include this post-processing module in order to offer as many options as possible to all of the integrated languages.

VI.3.1 Experiments in sentence selection

Nowadays, more and more resources are being made electronically available and, as a consequence, there is an ever-increasing number of sources for automatically generating questions. However, in the process of learning a language, the quality of the exercise is a matter of importance. In fact, it is obvious that the suitability of any item depends, among other features, on the source which is selected.

One way of predicting the best candidate sentences could be to base the selection on some measures of readability, such as reading difficulty measures. However, readability measures are usually defined as document readability measures, and so they require a passage of more than one sentence in order to compute the measure. Instead of focusing on such measures, we decided to follow some previously defined criteria by Kilgarriff *et al.* (2008). Although the proposed criteria were first established in order to provide good dictionary examples, the same criteria were then proposed within an automatic cloze generation task[4] (Smith *et al.*, 2009). In this study the following restrictions were proposed: (a) to give preference to sentences containing between 10 and 25 words; (b) to give preference to sentences in which the key term is in the main clause; (c) to penalise sentences that include words that are not part of a list of the commonest 17,000 words or which are rare words; (d) to penalise sentences with pronouns and anaphors, together with those containing more than two capital letters, punctuation marks and non-alphanumeric characters.

The baseline system for our experiments was established taking into consideration the restrictions as regards sentence length[5] and the position of the topic. Furthermore, based on the idea of the list of the frequency of the words, we decided to study word frequency lists and the information given in the Web 1T 5-gram dataset.

Gold standard creation

The baseline system was used to establish the gold standard. In order to select sentences to be presented to the teachers, two constraints were defined and set as the baseline. The system only took into account sentences of a particular length (between 12 and 25 words), while considering as candidate

[4] In this study, cloze refers to MCQs.

[5] Some modifications were applied to Smith *et al.* (2009).

sentences those in which the topic, the verb, was part of the main clause. This last step was carried out thanks to the information obtained from the parser.

The system chose 83 questions at random from the general corpus and 62 from the specific corpus out of an initial 200. In order to establish the gold standard, the expert language teacher had to decide whether a given sentence was appropriate to be the stem of a question. The expert accepted 52.73% of the questions generated by the baseline system. If we take into account the source of the item, the acceptance rate was 67.07% for the general source corpus and 50.79% for the specific one. This gold standard was used in the following experiments.

The idea behind the following two experiments is to offer candidate sentences which contain vocabulary which is known by learners. Thus, the aim is to offer sentences which are not a distraction and do not make the vocabulary task more difficult. The first experiment studies the use of word frequency lists and the second one investigates how to exploit the Web 1T 5-gram dataset.

First experiment: BNC and General Service List frequency lists

The aim of this experiment was to determine the influence of the entries in two lists when choosing candidate sentences. For this reason, we studied a lemmatised frequency list of BNC described by Kilgarriff (1997) and the General Service List (GSL) (West, 1953). The BNC list has 6318 entries which occur 800 times in the whole BNC. The GSL is a set of 2000 words which were selected to be of the greatest "general service" to learners of English. The BNC frequency list can be seen as a collection of the most common words, and although in the GSL the entries are not the most common words, the frequency was one of the criteria which were utilised when defining the list.

As the aim of this experiment is to offer sentences made up of vocabulary which is known by learners, we first established the acceptance threshold for the BNC and GSL lists, that is, the minimum number of words a sentence should have from the lists to be considered as a candidate sentence. The threshold values were established empirically from both corpora before the questions were generated.

In order to do so, we compared the results from the manual evaluation (the gold standard) with the automatically generated questions for each list.

Table VI.4 summarises the recall and precision level of each list. The precision measure expresses the number of valid items among those which were detected. Recall shows the number of valid items among all of the sentences that were considered as valid. The row entitled BNC takes into account the first 2000 entries, and BNC_ALL is the entire list. We made this distinction in order to see whether the amount of words affects the results, and to make the results more comparable to the GSL.

	GSL	BNC	BNC_ALL
Threshold	>= 0.6	>=0.65	>=0.75
Precision	0.63	0.62	0.61
Recall	0.77	0.79	0.86

Table VI.4: Precision and recall for all of the items

As Table VI.4 shows, for every 10 candidate sentences, six were valid, regardless of the list used. In fact, there were no significant differences between the lists. However, if we divide the results based on the source corpus, the results vary. Table VI.5 presents these results.

	General corpus		Specific corpus	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
GSL	74.58	80.00	47.92	71.88
BNC	71.21	85.45	47.83	68.75
BNC_ALL	68.49	90.91	50.00	78.13

Table VI.5: Precision and recall taking into account the source corpus

There is evidence that the proposed criteria worked better in the general scenario, as they obtained consistently better precision and recall values, presumably due to the fact that the vocabulary of the novel is more specialised and the overlap between the general corpus and the frequency list is also greater.

In order to see if it is more appropriate to make use of sentences from a general corpus, we looked at the Web 1T 5-gram dataset. The second experiment explores the option of offering the most common use of the verbs.

Second experiment: The Web 1T 5-gram dataset

Offering common examples of usage can be as important as providing sentences with known words. When dealing with vocabulary, teachers usually work with the most common examples of a word, i.e., they first present the meaning of the word in its most frequent collocations. Thus, we decided to define an experiment in order to find the most common collocations of the verbs from the AWL.

A preprocess was defined in order to collect the occurrences of each verb from the AWL. In order to obtain all of the occurrences, the system made use of three different patterns:

- (v w5 w6 w7 w8): the verb is the first word;
- (w3 w4 v w5 w6): the verb is in the middle;
- (w1 w2 w3 w4 v): the verb is the last word.

The system then identified the occurrences of the patterns which share the same words. For example, given the verb **reveal**, and the occurrences of the patterns **"UV light revealed nothing out"**, **"revealed nothing out of the"**, **"but the UV light revealed"**, and **"revealed something out of"** and their counts, the system only related the counts of the first three patterns. The reason for this is that they can be part of the same sentence: **"UV light revealed nothing out"** shares the words *"UV light"* with the pattern **"but the UV light revealed"** and the words *"nothing out"* with the pattern **"revealed nothing out of the."**

After that, the most common occurrences for the given verb tense and person were counted. Therefore, in this experiment, the defined criterion considered a sentence as a candidate if its three patterns had a minimum count which was established empirically.

	Precision (%)	Recall (%)
All items	65.22	34.48
General corpus	81.48	40.00
Specific corpus	42.11	25.00

Table VI.6: Precision and recall taking into account the source corpus

Comparing the obtained results with the gold standard, the degree of precision is 65.22% (see Table VI.6), i.e., more than six out of 10 sentences were valid, but the rate of recall is quite low (34.48%), although this is not a critical aspect in this scenario, as the number of available sentences is very high. With regard to the results divided by the source corpus, the results vary as they did in the previous experiment.

Conclusions derived from the experiments

This last experiment was an initial attempt to take advantage of the Web 1T 5-gram dataset in the sentence selection task. In comparison to the frequency lists, this type of resource offers significant possibilities for exploitation and exploration. That is why a more detailed study of this resource could yield new results and better recall. Nonetheless, in both experiments, there is evidence that all of the proposed criteria work much better when producing items from a general corpus.

Both experiments aimed to select good-quality candidate sentences to form part of items. For this reason, we have presented various experiments and evaluated the obtained results with a gold standard. In order to obtain the gold standard, EFL expert teacher made use of the post-editing environment presented in section IV.2.3 and improved in section V.3.1.2. The evaluation process confirms the appropriateness of the improved version of the environment. The expert mainly used the previously established criteria as the reasons for discarding the stems and, in few cases, the expert needed of the "other reasons" option. Thus, this evaluation confirms the appropriateness of the established criteria for rejecting the stems, independent of the target language of the items.

To sum up, this type of approach offers to teachers the chance of using real sentences, extracted from real corpora, as part of their curricula. In addition, we believe that the results as regards the Web 1T 5-gram dataset have opened up a promising research line in this field.

Nonetheless, the idea of using the words that are part of the source sentence as a criterion for selecting better candidate sentences could also be applied in the distractor generation task. The following section explores this idea.

VI.3.2 Distractor generation and selection

As previously mentioned, one of the novel aspects of this scenario is the ability of the system to automatically generate verbs as distractors which are similar to the key. For this reason, all of the verbs which are part of the AWL were considered as candidate distractors. As we have explained in chapter III, the AWL is divided into mutually exclusive sub-lists, and experts use the first five sub-lists at the B2 level and the rest at the C1 level.

Gold standard creation

The baseline system selected the candidate distractors on the basis of semantic similarity between verbs according to the distributional data (based on the BNC), that is, the verbs which were most similar to the key were selected as candidate distractors. The distractors were compared for similarity, employing the **information radius** measure. Furthermore, all candidate distractors had to match the target verb in terms of transitiveness/intransitiveness, tense and person.

The expert teacher marked 94.07% of the distractors which were part of the accepted questions as valid. This manual evaluation was carried out with 200 questions and the results obtained were set up as the gold standard. The gold standard does not vary significantly from one source corpus to the next.

Use of the sub-lists

In order to establish whether or not there was any difference in terms of appropriateness when verbs from the previous sub-lists were included as candidate distractors, we divided the distractors evaluated by the teacher according to their sub-lists. The results show that the use of more data does not result in an inferior performance. Thus, the use of verbs from previous AWL sublists was set up as a parameter of the system.

Language model from the Web 1T 5-gram dataset

In addition to offering distractors which are similar to the key in a ranked order, we also studied whether the context could play a beneficial role in the selection of distractors.

Various attempts to use context when deciding whether or not a candidate distractor is truly a distractor have already been made. Sumita *et al.* (2005)

proposed searching the Web in order to find any examples of use; in the event that a coincidence exists, the distractor is rejected. In our opinion, the method proposed by Sumita *et al.* (2005) is too restrictive, as it discards candidates as a result of just one occurrence on the web and the Web contains errors.

Smith *et al.* (2008) searched for candidate distractors and then selected sentences in which the correct answer and the distractor are mutually exclusive. In contrast, our system selects the sentence first and then generates the distractors.

The Web 1T 5-gram dataset was used to obtain a language model which predicted the probability of the occurrence of a word sequence. For this purpose, we used the smoothing method proposed by Kneser and Ney (1995), and we experimented with a 3-gram language model and a 5-gram language model. Our aim was to compare the probability of the n-gram containing the correct answer with the n-grams containing the candidate distractors.

As a first step, 40 MCQs from tests from CAE exams were analysed. Each question was composed of a correct answer and three distractors, and the topic was the semantics of verbs. We looked at the probability of the occurrence of the word sequences with the correct answers and distractors, expecting a higher probability of the sequences containing the correct answer.

The data show that, for the 3-gram model, the probability is lower for 86% of the distractors compared with the correct answers. Moreover, the probability of 80% of the distractors is smaller than a sixth of the correct 3-gram probability. In the case of the 5-gram model, the probability of 90% of the distractors is lower than a tenth of the correct 5-gram probability, but some of the obtained probabilities were almost zero. This is why, the 3-gram language model was selected as the criterion for selecting the distractors.

Thus, after analysing both language models with questions collected from preparatory tests of the CAE exam, a new heuristic rule was defined: a candidate distractor is confirmed as a distractor if its probability is lower than a sixth of the correct 3-gram probability and is greater than zero. This difference in probability was established empirically.

Once the heuristic was defined, we compared the results with the gold standard. The obtained precision level was 94.30% and recall 37.84%.

The heuristic offers a way to select different distractors, taking into account the context of the candidate sentence. Otherwise, the generated distractors would always be the same for each verb.

As a result of this set of experiments, two new heuristics were defined

type	**function**	**method**
semantic	similarity_measure(language,verb,language_model)	corpus_based(method)

Table VI.7: Representation of heuristics in the question model

and integrated into ArikIturri. Table VI.7 displays the way in which they are represented in the question model. The defined function contains three parameters. As the function was defined in order to be as general as possible, the *language* parameter established the language of the item. The *verb* parameter contains the key verb and finally the *language_model* Boolean parameter specifies whether any language model has to be used in the generation process. This type of general function allows a set of heuristics to be grouped into the same concept.

To sum up, the set of experiments presented in this section proves the multilinguality of the system by means of generating English verb tests. In addition, ArikIturri adds semantic information to grammatical information when generating distractors. More specifically, the way to apply distributional similarity measures has been studied and evaluated with expert teachers. The following section explores the possibility of employing semantic information in the generation of Basque items.

VI.4 MCQs for Basque science tests

As in the experiments presented in section VI.3, the aim of these new experiments is to propose distractors that correspond to the vocabulary studied as part of students' curricula. These experiments are focused on Basque scientific vocabulary within the science and technology domain in which the *distractor generation* module applies semantic relatedness measures in order to obtain the distractors.

The objective is to offer experts a tool to help them to create didactic resources. Human experts identified the meaningful terms (i.e., words) in an article,[6] which were used as the blanks in the MCQs. Then, the sys-

[6] These articles came from a website that provides current and up-to-date information on science and technology in Basque: http://www.zientzia.net.

tem applied semantic relatedness measures and used different resources such as corpora, ontologies and graph-based methods in the process of generating distractors. The aim of this section is to study different methods of automatically generating distractors of a high quality, i.e., distractors that correspond to the vocabulary studied by students as part of the curriculum.

Examples VI.4.1, VI.4.2 and VI.4.3 present three real items which were created automatically and which form part of a test.[7] The source of these items is a text relating to the *Earth*, the main aim of which is to explain the different methods "to cool" the *Earth*. Each example corresponds to one of the types of key analysed during the experiments. As we will explain in section VI.4.3, the analysis of the results was carried out taking into account the keys' PoS and semantic features. More specifically, the results distinguish between monosemous nouns, polysemous nouns and verbs.

Example VI.4.1 (Example of a monosemous key)
Itsasoko planktonak, esaterako, CO_2 *asko "irensten" du ...4... egitean. (The plankton in the sea, for instance, "eat" a lot of* CO_2 *when carrying out ...4...).*

a. fotosintesia (photosynthesis)	*c. korala (coral)*
b. izakia (being, creature)	*d. itsaskia (seafood)*

In example VI.4.1, the key fotosintesia (photosynthesis) is a monosemous noun which is specific to the domain of science and technology.

Example VI.4.2 (Example of a polysemous key)
East Angliako Unibertsitateko Tim Lenton irakasleak esan duenez, "geoigeniaritzak bakarrik ezin du arazoa konpondu, baina proposatutako mekanismo batzuk berotegi-efektuko gas-emisioen murriztearen ...19... izan daitezke". (As professor Tim Lenton from the University of East Anglia has said, "the geoengineering can not solve the problem on its own, but some of the proposed mechanisms can be the ...19... in the reduction of greenhouse gas emissions.")

a. osagarriak (complements)	*c. igorpenak (sendings)*
b. prozesuak (processes)	*d. ondorioak (conclusions)*

In example VI.4.2, the key osagarriak (complements) is a polysemous noun which is not only related to science and technology, but also to other domains.

[7] The distractors in these particular examples were obtained by applying heuristic 6, as explained in section VI.4.2.

Example VI.4.3 (Example of a verb key)
Bada, halaber, martxan dagoen beste teknika bat: karbonoa ...9... . (There is also another technique in use: ...9... the carbon.)

a. bahitzea (to abduct) *c. ehorztea (to cover)*
b. tratatzea (to treat) *d. jartzea (to place)*

In example VI.4.3, the key bahitzea (to abduct) is a verb and is also a term which is not only related to science and technology.

We did not give the expert any guidelines focusing on these experiments. The expert had to mark the most meaningful terms in the articles in order to test the students' scientific vocabulary. As the expert proceeded as usual when creating this type of exercise, not only domain-specific terms were marked, but also more general target terms.

From now on, these examples will be used to explain the different methods and results.

VI.4.1 Design of the scenario

As the aim was the generation and evaluation of items, some qualitative and quantitative analyses were conducted. More specifically, the aim was to identify problematic items. As previously mentioned, the qualitative analysis was based on expert knowledge, whereas the quantitative analysis was conducted after the items had been given to OSE students. A scenario was created in order to conduct the analyses. We designed experiments in which most of the external factors which could have an influence on the evaluation process were controlled. Thus, the process of generating and analysing the MCQs consisted of the following steps (Aldabe and Maritxalar, 2010):

1. Selection of the texts: experts in the generation of didactic resources select the texts for a specific domain, taking into account the level of the learners and the length of the texts;

2. Marking the blanks: the terms to be considered as keys have to be relevant within the text in order for them to be removed. The target terms have to be appropriate concepts for OSE students. The marking is carried out manually;

3. Generation of distractors: for each stem and key selected in the previous step, distractors are generated;

4. Choosing the distractors: experts have to verify that the automatically generated distractors cannot fit the blank;

5. Evaluation with learners: each learner reads the MCQs embedded in a text and chooses the correct answer from among four options;

6. Item analysis: based on the learners' responses, an item analysis is carried out in order to measure the quality of the distractors.

Blanks: One expert who works on the generation of learning materials was asked to mark between 15 and 20 suitable terms in four texts in order to create MCQs. The main topics of these texts were: *Continent*; the *Earth*; *Bats* and the *Arctic* respectively. The aim of the experiment was to evaluate the quality of the distractors in a real situation; that is why the blanks were marked manually. The expert did not follow any guidelines focusing on our experiment, but carried out the marking based on his experience. The blanks obtained were suitable in terms of their appropriateness for the science domain and the stems. In total, 94 blanks were obtained. As we did not give the expert any extra information for the marking process, the expert marked as keys nouns, verbs, adjectives and adverbs. However, from a computational point of view, our study aims to generate nouns and verbs. In total, 69.14% of the obtained blanks were nouns and 15.95% were verbs. This shows that the idea of working with nouns and verbs makes sense in a real situation. In total, 65 blanks were obtained: 17 in the text relating to *Continent*; 19 in the text relating to the *Earth*; 15 in the case of the text about *Bats* and 14 for the text on the *Arctic*.

Distractors: The distractors were generated automatically for each blank and method (cf., section VI.4.2). In the case of the nouns, five different methods were applied, while in the case of the verbs, three methods were applied. As this was a completely automatic process, it was not possible to provide learners with these distractors without supervision. Therefore, once the distractors were generated, the expert checked them. For each question and method, we provided the expert with the first four candidate distractors,[8] and the expert had to reject those distractors which could be correct answers.[9] In all cases, three valid distractors were obtained. Only 1.31% of the

[8] We had already decided to reject the items which had fewer than three appropriate distractors.

[9] In this task, the expert should not have to evaluate the quality of the distractors, but their correctness.

distractors could be a suitable key and 2.96% were rejected as dubious.

Schools and learners: A total of 18 different schools took part in the experiments. The exercise was presented to the learners as a test and the teachers were not familiar with the articles until they handed the test out to their students. In total, 951 OSE learners (second grade) participated in the evaluation. They had a maximum of 30 minutes to read and complete the test. The test was carried out on paper in order to avoid any noise. In total, 890 of the learners completed the test and their results were used to analyse the items. After finishing the test, an external supervisor collected the results of the exercise in situ.

It is important to analyse the item responses in a quantitative way, because this type of analysis provides, among other things, descriptions of item characteristics and test score properties. As clarified in section IV.2.2, we explored item difficulty, item discrimination and the evaluation of distractors based on CTT. As a consequence of this analysis, it is possible to identify some behaviours based on which certain actions can be performed.

For instance, based on the results of example VI.4.1, it is possible to conclude that the presented item is easy due to the high number of students who answered it correctly from among the different groups. More specifically, in all of the groups in which the test was conducted, students from different schools were able to answer this item correctly, independent of the generated distractors.

In the case of example VI.4.2, the results show that this was found to be a difficult item in all of the schools in which this text was tested. When analysing the selected options, we observed that the students did not tend to choose a particular distractor, but any of them. This kind of item can be appropriate for MCQs in order to motivate advanced students to improve their learning process. The detection of this kind of specific MCQs is interesting because they could be a dynamic addition into a hypothetical technology enhancement learning system (TELS), depending on the learner's needs.

Finally, as regards example VI.4.3, the item is difficult because many students were unable to answer it correctly. However, there is an important difference between example VI.4.2 and this example. In this case, most of the students chose the same distractor: tratatu (to treat), as the answer to the question. Therefore, this distractor in this MCQ should be revised or rejected before using it to test learners' knowledge.

VI.4.2 Generation approaches

Minority languages such as Basque lack some of the resources which are fundamental for measuring similarities. Although the aim of our work is to study the two main approaches to measuring semantic relatedness, we encountered some restrictions. First of all, we had to limit our study to nouns and verbs due to the aforementioned lack of resources. Second, as regards distributional similarity measures, the context to be represented had to be restricted to certain vector spaces. By the time the experiments were defined, it was not possible to obtain a modifier space for Basque words. The EPEC corpus (Aduriz *et al.*, 2006) was the only available Basque corpus which contained the necessary information on grammatical dependency in order to obtain noun-verb pairs. Nonetheless, the corpus was not big enough for our purposes. Finally, with regard to approaches based on knowledge resources, as the verbs in the Basque WordNet still need manual revision, the graph-based approach was only applicable for nouns.

We have implemented the corpus-based approach, the graph-based approach and a combination of both approaches. As it was not possible to apply the graph-based and the combined approaches to all of the keys of the items, we considered that the main approach for these domain-specific tests would be the corpus-based approach.

VI.4.2.1 Corpus-based approaches

As previously mentioned, our system deploys LSA in order to compute similarities among words. In order to build a VSM, ArikIturri makes use of Infomap software (Dorow and Widdows, 2003). This software uses a variant of LSA to learn vectors representing the meanings of words in a vector space known as WordSpace. As the MCQs we worked with were focused on the science domain, we made use of the *ZT corpus* because it contains texts relating to science and technology (cf., section IV.1.2.3). The software indexes the documents in the specialised corpus and performs word-to-word semantic similarity computations based on the resulting model. As a result, the system extracts the words that best match a query according to the model. Among the various options, we set the sentence as the context of the query and the words retrieved by the model as the starting point from which to generate the distractors. Thus, starting from this idea, different variants of the corpus-based method were defined.

LSA-based: The method offers as candidate distractors the first words of the output which are not part of the sentence and which match the same PoS.

LSA-based & specialised dictionary: This method combines the information offered by the model and the entries of the encyclopedic dictionary of science and technology for Basque (cf., IV.1.3.1). The 23,000 basic concepts relating to science and technology and divided into 50 different subjects are consulted in this approach. More specifically, based on the candidate distractors generated by the LSA method, the system searches the dictionary for the lemmas of the key and the distractors. If there is an entry for all of them, the candidate distractors which share the subject with the key in the encyclopedic dictionary are given preference. If not, the candidate distractors with an entry in the dictionary take preference in the selection process. In addition, those candidates which share any semantic characteristics with the key are preferred as suitable distractors.

LSA-based & ontology & morphology: One of the constraints defined by this method is the necessity of avoiding the possibility of students guessing the correct choice by discarding some options using their semantic or morphological information. In the first step, the system provides InfoMap with the entire sentence in which the key appears. The system offers the first words of the output which are not part of the sentence and which match the same PoS as candidate distractors.

For instance, with the stem Istripua izan ondoren, ... sendatu ninduen (After the accident, ... cured me) and the key medikuak (the doctor), the system proposes as a candidate distractor the word ospitalak (the hospital). Both words are related and belong to the same specific domain. However, learners could discard ospitalak as the answer to the question because they know that the correct option has to be a person in the given sentence. The system tries to avoid this kind of guessing by means of semantic information. Therefore, by applying this method, the system would not offer ospitalak as a candidate distractor.

In order to do so, in a second step, the system proposes only those candidates which share at least one semantic characteristic with the key. For this purpose, the system always tries to find the entries in the monolingual dictionary which look at the semantic features of common nouns obtained with the semiautomatic method (cf., section IV.1.3.1). Thus, if the key and the candidate distractor share any semantic features, the candidate distractor is proposed; if not, the system searches the characteristics in the MCR, which works with synsets (cf., section IV.1.3.2). In order to exploit the properties of the MCR, the system takes into account all of the synsets of the words and it decides whether or not they share any characteristics. Therefore, if a candidate distractor and the key share any of characteristics specified by the Top Concept Ontology, the WordNet Domains or SUMO, the candidate distractor is proposed.

Working with all of the senses of the words may yield invalid distractors in terms of semantics. Moreover, there are some cases in which two words share a semantic characteristic induced from the MCR, but in which the distractor would not be suitable because of its morphosyntactic features. For instance, while the lemma ospital (hospital) and the morpheme -ko form the word ospitaleko (of the hospital), it is not possible to combine the lemma mediku (doctor) with the suffix -ko, as -ko is only used to express the locative genitive case with inanimate words.

In the last step, the method looks at the morphosyntactic features of the candidate distractors. As the input text has previously been analysed by the morphosyntactic analyser, the system distinguishes the lemma and the morphemes of the key. It identifies the case marker of the key and it generates the corresponding inflected word for each candidate distractor, using as a basis the lemma of the distractor and the suffix of the key. Once distractors are generated with their corresponding forms, the system searches for any occurrence of the new inflected word in the *Euskaldunon Egunkaria* corpus (cf., section IV.1.2.1). If it occurs, the candidate distractor is selected.

VI.4.2.2 Graph-based approaches

The graph-based method regarding the distractor generation task is defined in four steps: first, a list of candidate distractors is obtained from WordNet. If the key is monosemous, all of its siblings are obtained as candidate distractors. In contrast, if the word has more than one meaning, the graph-based method is applied in order to obtain its most likely sense, and then its siblings are obtained. If it does not have siblings, the hyponyms of the key are considered as candidate distractors. Second, the personalised PageRank vector is obtained for the given context and the key. Third, the personalised PageRank vectors are obtained for 20 candidate distractors in the given context. Finally, the similarities among the vectors computed by the dot product are measured and a list of reordered candidate distractors is obtained.

VI.4.2.3 Combination of the corpus- and graph-based approaches

This method is a combination of the corpus-based and graph-based approaches to measuring the similarities. In this approach, the system computes similarity in two steps. First, it selects the candidate distractors based on the LSA model, and then the graph-based structure is used to refine the selection. The method is defined as follows: first, the system obtains a ranked list of candidate distractors using InfoMap. Second, the personalised PageRank vector is obtained for the context and the key. Third, the system applies the graph-based method for 20 candidates in the given context, obtaining each a personalised PageRank vector for each one. Finally, the similarities between the vectors computed by the dot product are measured and the candidate distractors are reordered.

VI.4.2.4 Heuristics

All of the different methods presented here were applied taking into account the features of the given keys. Thus, based on the PoS and the semantic features of the keys, we have defined different heuristics. These heuristics (Table VI.8) are basically combinations of the previously explained methods.

	Nouns in WordNet		Nouns not in WordNet	Verbs
	Monosemous	Polysemous		
Heuristic 1	LSA-based	LSA-based	LSA-based	LSA-based
Heuristic 2	LSA + O + M	LSA + O + M	LSA + O + M	LSA + O + M
Heuristic 3	LSA + Dict.	LSA + Dict.	LSA + Dict.	LSA + Dict.
Heuristic 4	Graph-based	Graph-based	LSA + O + M	LSA + O + M
Heuristic 5	Combination	Combination	LSA + O + M	LSA + O + M
Heuristic 6	Graph-based	Combination	LSA + O + M	LSA + O + M

Table VI.8: Heuristics (Legend: O: Ontology; M: Morphology)

As has been previously explained, the set of defined heuristics is represented in our question model in a particular way. Table VI.9 presents this specification.

The methods that were used to define the set of experiments are: corpus_based; graph_based; and combination. All of them have a *method* parameter which established the method to be used (lsa or ukb). In addition, the corpus_based method requires the specification of the use of additional information as part of the selection of candidate distractors.

Type	**Function**	**Method**
semantic	similarity_measure(language,verb,language_model)	corpus_based(method,sem_and_morph,dict)
		graph_based(method)
		combination(method1,method2)

Table VI.9: Representation of heuristics in the question model

In general, when the key is a verb, the only way to generate distractors is to apply a corpus-based method because, as has been previously stated, the verbs in the Basque WordNet still need manual revision. However, when the key is a noun, the system can apply different approaches. We have tested the heuristics according to the availability of the nouns in the Basque WordNet and the information that the WordNet gives about polysemy. In all of the heuristics, a generation process is applied in order to supply the distractors with the same inflected form as the key.

For instance, as the key fotosintesia (photosynthesis) of example VI.4.1 is a monosemous noun that appears in WordNet,[10] the following five methods were applied: the **LSA-based** method; the **LSA-based & specialised dictionary** method; the **LSA-based & ontology & morphology** method; the **graph-based** method and the method that **combines** the corpus-based and graph-based approaches. Table VI.10 presents the resulting distractors when the different methods were applied.

Method	Distractor 1	Distractor 2	Distractor 3
LSA-based	alga (seaweed)	oxigenoa (oxygen)	ura (water)
LSA + spec. dictionary	iraizketa (emission)	mantenu-gaia (maintenance-substance)	biziduna (organism)
LSA + ont. + morph.	izakia (creature)	corala (coral)	itsaskia (seafood)
Graph-based	laburbilketa (synthesis)	katalisia (catalysis)	higadura (erosion)
Combination	izakia (creature)	korala (coral)	itsaskia (seafood)

Table VI.10: Candidate distractors for the key fotosintesia (photosynthesis)

[10] 09705163n photosynthesis: synthesis of compounds with the aid of radiant energy (especially in plants).

In the case of example VI.4.2, for the key osagarriak, the aforementioned five methods were applied, thereby obtaining the distractors displayed in Table VI.11.

Method	Distractor 1	Distractor 2	Distractor 3
LSA-based	prozesuak (processes)	erabilgarritasunak (utilities)	eraginkortasunak (effectivities)
LSA + spec. dictionary	errausketak (incinerations)	konpostajeak (composting)	biogasak (bio-gases)
LSA + ont. + morph.	prozesuak (processes)	erabilgarritasunak (utilities)	eraginkortasunak (effectivities)
Graph-based	harreman-matematikoak (mathematical relations)	negozio-harremanak (business relations)	konparazioak (comparisons)
Combination	prozesuak (processes)	igorpenak (sendings)	ondorioak (conclusions)

Table VI.11: Candidate distractors for the key osagarriak (complement)

Regarding the variability of distractors, when five different methods are applied, a maximum of 15 different candidate distractors can be obtained (three for each method). In example VI.4.1, 12 different distractors are obtained, meaning that three distractors are generated twice.[11] Similar behaviour can be found in example VI.4.2, as 11 different distractors are obtained. For the key osagarriak (complement), the candidate distractor prozesuak (processes) is created using three methods, and the **LSA-based** and **LSA-based & ontology & morphology** methods create the candidate distractors erabilgarritasunak (utilities) and eraginkortasunak (effectivities).[12]

Finally, as concerns example VI.4.3, three different methods have been applied because the key bahitzea (to abduct) is a verb. Table VI.12 presents the obtained distractors.

In this case, the distractor tratatzea (to treat) is generated by the three methods and the distractor ehortzea (to cover) is obtained when applying the **LSA-based** and **LSA-based & ontology & morphology** methods are applied. Therefore, the percentage of different distractors is 66.66 % (six distractors out of nine) for the key bahitzea (to abduct).

[11] In this particular example, the **LSA-based & ontology & morphology** and the **Combined** methods create the same three distractors.

[12] In this particular example, both methods create the same three distractors.

Method	Distractor 1	Distractor 2	Distractor 3
LSA-based	tratatzea (to treat)	pasaraztea (force to pass)	ehortzea (to cover)
LSA + spec. dictionary	tratatzea (to treat)	konbinatzea (to combine)	erauztea (to extract)
LSA + ont. + morph.	tratatzea (to treat)	ehortzea (to cover)	jartzea (to place)

Table VI.12: Candidate distractors for the key bahitzea (to abduct)

From the analysis of the results obtained for the 65 keys (19 in the *Earth* text, 17 in the *Continent* text, 15 in the *Bats* text and 14 in the *Arctic* text), three main ideas were formed:

1. The percentage of different distractors is not related to the PoS of the corresponding key in the MCQ;

2. In the case of polysemous nouns, there is not a big difference between the distractors generated with the LSA method and those following the application of some semantic and morphosyntactic criteria to the output of the LSA. For example, 72.72% of the distractors are identical in the *Continent* text and 73.33% in the *Earth* text. Therefore, applying the LSA method only can be a good decision when there is a lack of semantic resources and lexical databases;

3. Obtaining a high number of different distractors, e.g., 80% in the case of the key osagarriak (complements), shows that a quantitative study designed to compare the tests generated by different heuristics makes sense.

VI.4.3 Evaluation

One of the particularities of this set of experiments is that ArikIturri generates Basque MCQs embedded in an entire text. In contrast, almost all of the literature regarding the automatic generation of distractors based on NLP methods is focused on English and isolated MCQs.

The scenario which has been designed allows the quantitative evaluation of the generated tests (cf., section VI.4.3.1). In addition, there are some interesting analyses that could help in understanding the generation approaches as well as the influence of the resources: qualitative and quantitative evaluation (cf., section VI.4.3.2); manual evaluation (cf., section VI.4.3.3) and the analysis of the influence of the occurrences of the keys (cf., section VI.4.3.4). All of these analyses are based on the results obtained from 951 students from 18 different schools. A total of 890 of the learners completed the test and their results were used to analyse the items. As explained in section IV.2, we have studied item difficulty, item discrimination and the analysis of distractors based on CTT.

In this work, we marked an item as easy if more than 90% of the students answered it correctly. On the other hand, an item was defined as difficult when less than 30% of the students chose the correct answer. The desired value of item difficulty is 0.5, and the number of easy and difficult items should not be high.

The results regarding item discrimination and distractors evaluation were obtained based on the low-scoring and high-scoring students. The top third of the students with the highest marks in the given test were considered to be the high-scoring group, while the bottom third of the class (with the lowest marks) were considered as the low-scoring group.

With regard to item discrimination, a positive value is desirable because it is an indication that the item was answered correctly by high-scoring students and incorrectly by low-scoring students.

Finally, as regards the analysis of the distractors, two results are interesting. On the one hand, the high selection rate of each distractor. On the other hand, the identification of distractors that work properly, that is, distractors that attract more students from the low-scoring group than from the high-scoring one.

VI.4.3.1 Quantitative analysis

As one of our aims is a comparison of the methods used to generate distractors, the experiments presented here are focused on statistical analysis. For statistical computing, we used R, a free software environment.[13] Due to the number of students, it was not possible to test the four texts using all of the heuristics. In addition, the analysis of the item difficulty and item discrimination values for different sets of students led us to set 30 as a reasonable sample size for comparing the different approaches. In total, we analysed the

[13] http://www.r-project.org

Earth	Test 1	Heuristic 1	Continent	Test 10	Heuristic 1
	Test 2	Heuristic 2		Test 11	Heuristic 2
	Test 3	Heuristic 3		Test 12	Heuristic 3
	Test 4	Heuristic 4		Test 13	Heuristic 4
	Test 5	Heuristic 5		Test 14	Heuristic 5
	Test 6	Heuristic 6		Test 15	Heuristic 6
Bat	Test 7	Heuristic 2	Arctic	Test 16	Heuristic 2
	Test 8	Heuristic 4		Test 17	Heuristic 6
	Test 9	Heuristic 6			

Table VI.13: The analysed tests

item difficulty and discrimination of 17 tests, as Table VI.13 summarises.

In addition, the analysis of the results differentiated between the PoS and polysemy of the keys in order to analyse whether these features can have an influence over the results.

Looking at the blanks marked by the expert, we can see that the expert chose mainly polysemous nouns as the keys in three texts (subjects: *Earth, Continent* and *Bats*): polysemous nouns: 52.63%, 64.71% and 60.00% respectively; monosemous nouns: 31.58%, 17.65% and 33.33% respectively; and verbs: 15.79%, 17.65% and 6.67% respectively. In the fourth text (*Arctic*) the percentages changed: polysemous nouns: 35.71%; monosemous nouns: 35.71%; and verbs: 7.14%. Although there is a difference as regards the marked keys, the average percentage of monosemous and polysemous nouns was similar in all of the texts. It is important to clarify that all of the polysemous nouns chosen by the expert appear in WordNet; however, some of the monosemous nouns do not appear in WordNet.[14] From now on, the results regarding these monosemous nouns are presented separately with the tag NOT WN.

Table VI.14 presents the results as regards the item difficulty. Each row of Table VI.14 represents the item difficulty index average, together with the standard deviation. The desired value for the the item difficulty index is 0.5, and none of the heuristics obtain this average. In fact, the results vary from one text to another (from 0.64 to 0.75). Based on the PoS of the keys, all

[14] As we had previously acknowledged this possibility, the defined heuristics take into account this distinction.

Item difficulty					
Earth (19)	Monosemous (4)	Polysemous (10)	NOT WN (2)	Verbs (3)	Overall
Heuristic 1	**0.67 (0.14)**	0.75 (0.19)	0.61 (0.03)	0.47 (0.33)	0.67 (0.20)
Heuristic 2	0.71 (0.19)	0.76 (0.23)	0.66 (0.14)	0.57 (0.36)	0.71 (0.21)
Heuristic 3	0.68 (0.14)	0.71 (0.21)	**0.58 (0.22)**	0.56 (0.40)	0.66 (0.22)
Heuristic 4	0.72 (0.18)	**0.64 (0.26)**	0.72 (0.10)	0.45 (0.26)	**0.64 (0.23)**
Heuristic 5	0.68 (0.29)	0.73 (0.22)	0.68 (0.04)	0.52 (0.25)	0.68 (0.21)
Heuristic 6	0.72 (0.28)	0.67 (0.24)	0.81 (0.04)	**0.50 (0.28)**	0.67 (0.24)
Continent (17)	Monosemous (3)	Polysemous (11)	-	Verbs (3)	Overall
Heuristic 1	0.78 (0.17)	0.83 (0.19)	-	0.68 (0.15)	0.79 (0.18)
Heuristic 2	0.85 (0.13)	0.84 (0.16)	-	0.74 (0.11)	0.83 (0.15)
Heuristic 3	0.81 (0.09)	**0.73 (0.22)**	-	0.51 (0.30)	**0.70 (0.23)**
Heuristic 4	0.74 (0.11)	0.88 (0.06)	-	0.70 (0.07)	0.82 (0.1)
Heuristic 5	0.89 (0.06)	0.80 (0.17)	-	**0.50 (0.2)**	0.76 (0.22)
Heuristic 6	**0.73 (0.21)**	0.77 (0.16)	-	0.64 (0.27)	0.74 (0.18)
Bats (15)	Monosemous (3)	Polysemous (9)	NOT WN (2)	Verbs (1)	Overall
Heuristic 2	0.92 (0.02)	0.83 (0.27)	0.95 (0.00)	0.76	0.86 (0.22)
Heuristic 4	0.67 (0.29)	0.82 (0.13)	1.00 (0.00)	**0.68**	0.80 (0.18)
Heuristic 6	**0.67 (0.17)**	**0.71 (0.25)**	**0.92 (0.02)**	0.69	**0.73 (0.22)**
Arctic (14)	Monosemous (5)	Polysemous (5)	NOT WN (1)	Verb (3)	Overall
Heuristic 2	0.85 (0.16)	0.65 (0.30)	0.76	0.75 (0.24)	0.75 (0.23)
Heuristic 6	**0.71 (0.13)**	**0.55 (0.32)**	**0.71**	**0.72 (0.26)**	**0.66 (0.23)**

Table VI.14: Average item difficulty

of the heuristics obtain better results for verbs, and there are no significant differences between the monosemous and polysemous nouns.

Looking at each text, while Heuristic 6 obtains the best results on average (overall column in Table VI.14) in the *Bats* and *Arctic* texts, Heuristic 4 in the *Earth* text and Heuristic 3 in the *Continent* text attain slightly better difficulty values.

	Item Difficulty			
Earth - (19)	Easy	%	Difficult	%
Heuristic 1	**1**	5.26	1	5.26
Heuristic 2	4	21.05	2	10.53
Heuristic 3	**1**	5.26	1	5.26
Heuristic 4	2	10.53	2	10.53
Heuristic 5	5	26.32	**3**	15.79
Heuristic 6	3	15.79	2	10.53
Continent - (17)	Easy	%	Difficult	%
Heuristic 1	5	29.41	0	0.00
Heuristic 2	6	35.29	0	0.00
Heuristic 3	**3**	17.65	**2**	11.76
Heuristic 4	5	29.41	0	0.00
Heuristic 5	4	23.53	1	5.88
Heuristic 6	**3**	17.65	0	0.00
Bats - (15)	Easy	%	Difficult	%
Heuristic 2	10	66.67	**1**	6.67
Heuristic 4	6	40.00	0	0.00
Heuristic 6	**4**	26.67	**1**	6.67
Arctic - (14)	Easy	%	Difficult	%
Heuristic 2	4	28.57	1	7.14
Heuristic 6	**1**	7.14	**2**	14.29

Table VI.15: Number of easy and difficult items per test

Table VI.15 shows the results regarding item difficulty based on the number of easy and difficult items generated per heuristic and per text. A test should comprise some easy items in order to encourage low-scoring students and some difficult items in order to stimulate high-scoring students. Thus, a balance between both should be the ideal. In the case of automatically generated items, Heuristic 6 produced the lowest number of easy items for the *Bats*, *Arctic* and *Continent* texts. Heuristic 3 obtained the same number of easy items for the *Continent* text. The lowest number in the *Earth* text

	Item discrimination					
Earth (19)	Monosemous (4)	Polysemous (10)	NOT WN (2)	Verbs (3)	Overall	Neg.
Heuristic 1	0.04 (0.09)	0.12 (0.25)	0.18 (0.29)	0.04 (0.23)	0.10 (0.21)	6
Heuristic 2	0.36 (0.14)	0.39 (0.20)	0.32 (0.15)	0.25 (0.24)	0.30 (0.15)	1
Heuristic 3	0.48 (0.34)	0.09 (0.39)	0.32 (0)	0.19 (0.15)	0.22 (0.35)	5
Heuristic 4	0.18 (0.17)	0.01 (0.17)	-0.02 (0.05)	-0.01 (0.20)	0.04 (0.17)	8
Heuristic 5	0.43 (0.15)	0.42 (0.18)	0.30 (0.17)	0.39 (0.33)	0.41 (0.19)	**0**
Heuristic 6	0.19 (0.23)	0.13 (0.22)	0.22 (0.07)	0.17 (0.09)	0.16 (0.19)	3
Continent (17)	Monosemous (3)	Polysemous (11)	-	Verbs (3)	Overall	Neg.
Heuristic 1	0.37 (0.29)	0.24 (0.21)	-	0.02 (0.24)	0.22 (0.25)	3
Heuristic 2	0.23 (0.30)	0.27 (0.16)	-	0.24 (0.16)	0.26 (0.16)	**0**
Heuristic 3	0.15 (0.04)	0.22 (0.14)	-	0.30 (0.22)	0.22 (0.14)	1
Heuristic 4	0.49 (0.09)	0.43 (0.15)	-	0.35 (0.17)	0.43 (0.14)	**0**
Heuristic 5	0.30 (0.26)	0.33 (0.36)	-	0.31 (0.13)	0.33 (0.30)	2
Heuristic 6	0.58 (0.14)	0.24 (0.17)	-	0.34 (0.08)	0.31 (0.2)	**0**
Bats (15)	Monosemous (3)	Polysemous (9)	NOT WN (2)	Verbs (1)	Overall	Neg.
Heuristic 2	0.29 (0.22)	0.10 (0.21)	0.11 (0)	0.05	0.14 (0.20)	4
Heuristic 4	0.33 (0.16)	0.21 (0.18)	-	0.13	0.23 (0.17)	1
Heuristic 6	0.45 (0.16)	0.37 (0.26)	0.59 (0.04)	0.31	0.41 (0.22)	**0**
Arctic (14)	Monosemous (5)	Polysemous (5)	NOT WN (1)	Verb (3)	Overall	Neg.
Heuristic 2	0.26 (0.07)	0.11 (0.15)	0.04	0.20 (0.10)	0.17 (0.13)	**1**
Heuristic 6	0.22 (0.05)	0.18 (0.41)	0.12	0.08 (0.15)	0.17 (0.25)	2

Table VI.16: Average item discrimination

are obtained by means of Heuristics 3 and 1. Heuristic 2 tends to create the highest number of easy items. Overall, there is no method which specifically creates difficult items.

Table VI.16 presents the average item discrimination values for each heuristic and text. The numbers in parentheses denote the number of items per article and word type. With regard to the item discrimination values, the values in parentheses denote the standard deviations. All of the heuristics obtained a positive average value (overall column in Table VI.16), meaning that all of the heuristics gave the desired performance. There is no heuristic which stands out from the rest and there is no clear difference in the results based on the PoS.

The last column of Table VI.16 represents the negative discrimination values. In other words, it specifies the number of items which were answered correctly by a higher number of low-scoring students than high-scoring ones. All of the items generated by Heuristic 5 obtained positive discrimination values in the *Earth* text. No negative discrimination values were obtained by applying Heuristic 2, 4 and 6 to the *Continent* text. Heuristic 6 also attained only positive discrimination values for the *Bats* text. Finally, Heuristic 2 obtained the lowest negative discrimination values in the *Arctic* text with only one negative item. Heuristic 6 attained one more negative item than Heuristic 2 in the same text.

In order to identify the reasons for this course of action, we studied the option-by-option responses of the high-scoring and low-scoring groups. This study led us to evaluate the distractors themselves. For this purpose, we took into account two measures: the number of selected distractors and the positive discrimination value. The percentage of selected distractors was computed by dividing the number of selected distractors by the total number of distractors created by ArikIturri. The percentage of positive discrimination was calculated by dividing the number of distractors with positive discrimination by the total number of distractors. In this analysis, positive discrimination refers to those distractors that attracted more students from the low-scoring group than from the high-scoring one.

	Distractors analysis									
Earth	Monosemous (4)		Polysemous (10)		NOT WN (2)		Verb (3)		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Heuristic 1	83.33	50.00	70.00	53.33	**100.00**	**83.33**	66.67	33.33	75.44	52.63
Heuristic 2	**100.00**	**91.67**	66.67	60.00	83.33	**83.33**	55.56	44.44	73.68	**66.67**
Heuristic 3	83.33	75.00	63.33	36.67	83.33	66.67	77.78	55.56	71.93	50.88
Heuristic 4	75.00	58.33	66.67	43.33	83.33	50.00	66.67	33.33	70.18	45.61
Heuristic 5	66.67	66.67	73.33	**63.33**	83.33	66.67	**88.89**	**66.67**	75.44	64.91
Heuristic 6	83.33	50.00	**93.33**	50.00	66.67	**83.33**	**88.89**	**66.67**	**87.72**	56.14
Continent	Monosemous (3)		Polysemous (11)		-		Verb (3)		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Heuristic 1	66.67	66.67	39.39	33.33	-	-	**88.89**	44.44	52.94	41.18
Heuristic 2	44.44	44.44	48.48	45.45	-	-	44.44	44.44	47.06	45.10
Heuristic 3	66.67	66.67	57.58	48.48	-	-	77.78	55.56	62.75	52.94
Heuristic 4	**77.78**	**77.78**	66.67	**66.67**	-	-	**88.89**	**88.89**	**72.55**	**72.55**
Heuristic 5	44.44	33.33	**69.70**	63.64	-	-	66.67	55.56	64.71	56.86
Heuristic 6	66.67	66.67	63.64	63.64	-	-	77.78	66.67	66.67	64.71
Bats	Monosemous (3)		Polysemous (9)		NOT WN (2)		Verb (1)		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Heuristic 2	55.56	55.56	44.44	40.74	16.67	16.67	33.33	33.33	42.22	40.00
Heuristic 4	77.78	66.67	55.56	44.44	0.00	0.00	33.33	33.33	51.11	42.22
Heuristic 6	**100.00**	**100.00**	**81.48**	**77.78**	**66.67**	**66.67**	**100.00**	**66.67**	**84.44**	**80.00**
Arctic	Monosemous (5)		Polysemous (5)		NOT WN (1)		Verb (3)		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Heuristic 2	60.00	53.33	73.33	60.00	66.67	66.67	**77.78**	**66.67**	69.05	59.52
Heuristic 6	**93.33**	**73.33**	**86.67**	**73.33**	**100.00**	**100.00**	55.56	33.33	**83.33**	**66.67**

Table VI.17: Distractors analysis. (Diff. denotes the percentage of selected distractors; L > H denotes the percentage of positive discrimination)

Table VI.17 presents the percentage of selected distractors (Diff. columns) and the percentage of positive discrimination (L > H columns). Overall, Heuristic 6 produced distractors that attracted more students for the *Earth*, *Bats* and *Arctic* texts and Heuristic 4 for the *Continent* text. A near-identical performance obtained in relation to the percentage of items with positive discrimination. While Heuristic 6 obtained the highest number of distractors featuring positive discrimination in the *Bats* text (80%) and the *Arctic* text (66.67%), Heuristic 4 attained the best results for the *Continent* text (72.55%) and Heuristic 2 for the *Earth* text (66.67%).

It is worth noting that the percentage of items selected by only one or two students was fairly high for all of the methods. This is due to the fact that we used groups of 30 students per test, meaning that we were analysing the results of 10 students per low-scoring or high-scoring group.

The distractors featuring negative discrimination need to be revised by an expert because they confused high-scoring students more than low-scoring students. All of the heuristics created this type of item although the percentage is not very high. Looking at the distractors that discriminate in a negative way with a difference of at least two students, we can see that these problematic distractors are as follows: in the case of Heuristic 1, the **distractor** algak (seaweed) for the *key* planktonak (plankton) and erregaia (fuel) for the *key* CO_2-a (CO_2); Heuristic 3 created distractors which were directly rejected for the *keys* teknologia (technology), teknikak (techniques) and osagarriak (complements); these **distractors** were ingeniaritza (engineering), ikerketak (studies) and konpostajeak (composting) respectively. Two heuristics (4 and 5) generated the problematic **distractor** ehortzea (to cover) for the *key* verb bahitzea (to abduct). Finally, students tended to select the **distractor** ekoizpen-mailaren (level of production) created by Heuristic 6 instead of the corresponding *key* itsasoaren (sea).

Based on all of the results, we consider that the Heuristic 6 is the best overall strategy for producing distractors. The following section presents one experiment which focuses on the evaluation of the items in a real scenario. As this type of evaluation is expensive, we had to restrict the generation to one heuristic. Thus, the system generated distractors using Heuristic 6.

VI.4.3.2 Real scenario

Although in the previous section, we analysed the results of the tests in a quantitative way in order to find the best heuristic, in a real scenario, item analysis should be carried out in two steps: first, by giving the items to experts (qualitative analysis), and then by evaluating the supervised tests with students (quantitative analysis). This new evaluation aims to study the items in such a way. It is necessary to note that the potential end-users of ArikIturri are experts or teachers who are looking for items to test their students. Thus, this two-step process constitutes the way in which they would proceed in their everyday work. As previously mentioned, the system applied Heuristic 6 in order to generate the distractors and generated 10 distractors per item.

For the qualitative analysis, four tests were generated, one per text. In this first step, we gave the distractors to an expert, and the expert had to select the three most appropriate ones.

In the event that there were not three appropriate distractors, we asked him to generate distractors so that there would be three. In this way, we obtained four tests in which suitable distractors were selected or generated by the expert.

		Monosemous	Polysemous	NOT WN	Verbs	Total
Earth	#Distractors	12	30	6	9	57
	Manually	1 (8.33%)	10 (33.33%)	2 (33.33%)	1 (11.11%)	14 (24.56%)
Continent	#Distractors	9	33	-	9	51
	Manually	4 (44.44%)	11 (33.33%)	-	0 (0.00%)	15 (29.41%)
Bats	#Distractors	9	27	6	3	45
	Manually	1 (11.11%)	10 (37.04%)	0 (0.00%)	0 (0.00%)	11 (24.44%)
Arctic	#Distractors	15	15	3	9	42
	Manually	3 (20.00%)	3 (20.00%)	0 (0.00%)	0 (0.00%)	6 (14.29%)
	Total	45	105	15	30	195
	Manually	9 (20.00%)	34 (32.38%)	2 (13.33%)	1 (3.33%)	46 (23.59%)

Table VI.18: Number of manually created distractors

Table VI.18 presents the total number of distractors per PoS and per semantic features (#Distractors rows), together with the number of manually generated distractors (Manually rows). In three of the texts (*Earth*, *Continent* and *Bats*) the expert created a similar number of new distractors (between 25% and 30% of the total), because he considered that those which were being offered were not appropriate. However, the number of items which were completely replaced[15] was higher in the *Continent* text (23.53%) than in the other two (10.53% in the *Earth* text and 13.33% in the *Bats* text).

[15] That means that the three distractors which were generated automatically were not appropriate.

The *Arctic* text obtained the best results on average: just 14.29% of the distractors were created manually, corresponding to the items which were completely replaced (14.29%). If we split the distractors by their PoS, it is important to remark that, in the case of the verbs, only one of the automatically generated distractors was rejected by the expert (3.33%). In the case of monosemous and polysemous nouns, this percentage increased to 18.33% and 32.38% respectively.

Therefore, once the tests had been manually revised, four new tests were ready to be distributed to the students. Due to the number of students available, we had to limit the analysis to three of the revised tests and so we chose the tests relating to the *Earth*, *Continent* and *Bats* because they had the highest number of keys.

	Item difficulty				
	Monosemous	Polysemous	NOT WN	Verbs	Overall
Earth (19)					
Real scenario	0.67 (0.26)	0.49 (0.30)	0.73 (0.24)	0.43 (0.26)	0.55 (0.28)
Heuristic 6	0.72 (0.28)	0.67 (0.24)	0.81 (0.04)	0.50 (0.28)	0.67 (0.24)
Continent (17)					
Real scenario	0.60 (0.10)	0.64 (0.21)	-	0.52 (0.15)	0.62 (0.18)
Heuristic 6	0.73 (0.21)	0.77 (0.16)	-	0.64 (0.27)	0.74 (0.18)
Bats (15)					
Real scenario	0.76 (0.22)	0.64 (0.33)	0.95 (0.07)	0.97	0.73 (0.29)
Heuristic 6	0.67 (0.17)	0.71 (0.25)	0.92 (0.02)	0.69	0.73 (0.22)

Table VI.19: Comparison of item difficulty in a real scenario

Table VI.19 presents the results as regards item difficulty, comparing the results from the real scenario and the previously displayed results for Heuristic 6. Each value in the table represents the average item difficulty value, together with the standard deviation. If we compare these results with the ones obtained when conducting the tests without any supervision, the overall item difficulty value is closer to the desired value of 0.5. The only exception is when the experiments were conducted with the *Bats* text.

Table VI.20 compares the number of easy and difficult items per test. As previously mentioned, a balance between the number of easy and difficult items is desirable and in an automatic system, a high number of difficult items is preferable, to a high number of easy items. The number of easy items decreases in the *Earth* and *Continent* texts and, in contrast, increases

	Item difficulty			
Earth - (19 MCQ)	Easy	%	Difficult	%
Real scenario	2	10.53	6	31.58
Heuristic 6	3	15.79	2	10.53
Continent - (17 MCQ)	Easy	%	Difficult	%
Real scenario	0	0.00	1	5.88
Heuristic 6	3	17.65	0	0.00
Bats - (15 MCQ)	Easy	%	Difficult	%
Real scenario	6	40.00	2	13.33
Heuristic 6	4	26.67	1	6.67

Table VI.20: Comparison of the number of easy and difficult items per test

in the *Bats* text. As regards the number of difficult items, it increases for all of the texts when comparing the results of the real scenario with those of Heuristic 6.

	Item discrimination					
	Monosemous	Polysemous	NOT WN	Verbs	Overall	Neg.
Earth (19)						
Real scenario	0.09 (0.11)	0.17 (0.21)	0.22 (0.17)	0.25 (0.03)	0.17 (0.17)	3
Heuristic 6	0.19 (0.23)	0.13 (0.22)	0.22 (0.07)	0.17 (0.09)	0.16 (0.19)	3
Cont. (17)						
Real scenario	0.35 (0.09)	0.24 (0.16)	-	0.20 (0.22)	0.25 (0.16)	1
Heuristic 6	0.58 (0.14)	0.24 (0.17)	-	0.34 (0.08)	0.31 (0.2)	0
Bats (15)						
Real scenario	0.08 (0.09)	0.00 (0.26)	x	0.13	0.04 (0.21)	6
Heuristic 6	0.45 (0.16)	0.37 (0.26)	0.59 (0.04)	0.31	0.41 (0.22)	0

Table VI.21: Comparison of item discrimination in the real scenario

Table VI.21 presents a comparison of the item discrimination results of the real scenario and Heuristic 6. In the case of the *Earth* and *Continent* texts, there is no difference in the results. However, the number of items that discriminates negatively in the *Bats* text increases substantially. This type of item is problematic, and the items need to be revised. In order to see whether the distractors are the cause, we also analysed the distractors one-by-one.

Table VI.22 presents the analysis of the distractors. For each test, we analysed the number of selected distractors and the number of distractors

	Monosemous		Polysemous		NOT WN		Verb		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Earth (19)										
Real	83.33	66.67	93.33	73.33	66.67	66.67	88.89	77.78	87.72	71.93
Heuristic 6	83.33	50.00	93.33	50.00	66.67	83.33	88.89	66.67	87.72	56.14
Cont. (17)										
Real	66.67	66.67	84.85	72.73	-	-	100.00	55.56	84.31	68.63
Heuristic 6	66.67	66.67	63.64	63.64	-	-	77.78	66.67	66.67	64.71
Bats (15)										
Real	66.67	66.67	66.67	48.15	16.67	16.67	33.33	33.33	57.78	46.67
Heuristic 6	100.00	100.00	81.48	77.78	66.67	66.67	100.00	66.67	84.44	80.00

Table VI.22: Comparison of distractors analysis

that discriminated positively. As occurred with item discrimination, the percentage of selected distractors and positive discrimination increased in comparison with the results of Heuristic 6, with the exception of the *Bats* text.

With regard to the distractors which were selected by the expert when creating the test relating to the *Bats* text, the expert selected three out of a list of 10 distractors. The expert chose the distractors that he considered to be the most appropriate to test the students' knowledge. Obtaining fewer distractors with positive discrimination led us to conclude that, even for experts, the generation of distractors is a difficult task. This could explain the lower results obtained for the *Bats* text.

The expert needed almost the same amount of time to carry out this task as to generate all of the distractors manually. However, by means of automatic generation, experts have the chance to select distractors that, otherwise, they would never produce. In section VI.4.3.3, we present experiments in which the expert had to create distractors manually. Based on this information, for instance, for the *key* neguak (winter), the expert first generated as **distractors** udak (summer), udaberriak (spring) and udazkenak (autumn). In contrast, when the task was to select the most appropriate distractors, even though he had the same candidate distractors, instead of selecting udak (summer), he selected urtaro euritsuak (rainy season).

VI.4.3.3 Generation of distractors by hand

In addition to the previously explained analysis designed to measure the quality of the defined heuristics, another way is to compare them with manually generated tests. In order to see how good and real the heuristics are, one expert manually created distractors for each text. He had to find distractors which were semantically close to the key, that is, he had to base the creation on semantic similarity measures.

One of the first analyses shows that there are some manually generated distractors that the system also generates automatically. For example, in the *Earth* text, four distractors out of 12 are repeated when generating distractors for monosemous keys, and two distractors out of 30 are repeated when generating distractors for polysemous keys. In the other texts, the number of distractors that match is lower. Table VI.23 displays these generated distractors in detail.

Text	**Key**	**Distractor**
Earth	planktonak (plankton)	animaliek (animals)
	neguak (winter)	udak (summer)
		udazkenak (autumn)
		udaberriak (spring)
	teknologia (technology)	jakintza (knowledge)
	teknikak (techniques)	ikerketak (studies)
Continent	dentsitatea (density)	tenperatura (temperature)
Bats	arazo (problem)	ondorio (consequence)
Arctic	gasa (gas)	ikatza (coal)

Table VI.23: Manually and automatically generated distractors (matched)

Table VI.24 presents the average values of the manually built tests as regards the items' difficulty. These results and the following results should be seen as the upper bound for ArikIturri. Overall, the manual method is better than all of the automatic methods, but the real scenario obtains similar results in terms of item difficulty.

	Item difficulty				
Text	Monosemous	Polysemous	NOT WN	Verb	Overall
Earth	0.59 (0.31)	0.34 (0.23)	0.69 (0.24)	0.44 (0.23)	0.46 (0.26)
Continent	0.75 (0.17)	0.68 (0.24)	-	0.52 (0.28)	0.66 (0.23)
Bats	0.59 (0.27)	0.59 (0.28)	0.50 (0.10)	0.9	0.60 (0.25)
Arctic	0.70 (0.15)	0.63 (0.30)	0.8	0.78 (0.21)	0.70 (0.21)

Table VI.24: Item difficulty of manually generated items

Looking at the number of easy and difficult items in Table VI.25, the expert tended to create more balanced tests in terms of difficulty than the other options, including the real scenario. However, there is no such difference between the manual generation and the real scenario if we take into account item difficulty only. In fact, the results from the real scenario could be comparable to the manually created items with the exception of the *Bats* text.

	Item difficulty			
	Easy	%	Difficult	%
Earth - (19 MCQ)	0	0	7	36.84
Continent - (17 MCQ)	0	0.00	1	5.88
Bats - (15 MCQ)	1	6.67	1	6.67
Arctic - (14 MCQ)	3	21.43	1	7.14

Table VI.25: Number of easy and difficult items per manually generated test

Table VI.26 presents the obtained average values as regards item discrimination. Overall, the expert created more items with negative discrimination values. That is, the automatically generated tests have fewer negative items than the manually generated ones. In fact, the same behaviour was detected within the real scenario. This may be due to the fact that the expert tended to create more difficult items so that the distractors would be more attractive to the entire group of students.

	Item discrimination					
Text	Monosemous	Polysemous	NOT WN	Verb	Overall	Neg.
Earth	0.23 (0.24)	0.08 (0.21)	0.30 (0.14)	0.18 (0.20)	0.14 (0.21)	4
Continent	0.13 (0.06)	0.12 (0.20)	-	0.17 (0.20)	0.13 (0.21)	4
Bats	0.15 (0.10)	0.01 (0.16)	-0.02 (0.00)	0.26	0.05 (0.15)	5
Arctic	0.06 (0.11)	0.05 (0.19)	0.21	0.01 (0.19)	0.06 (0.15)	6

Table VI.26: Item discrimination of manually generated items

One more interesting result was obtained in our research as regards the *Earth* and *Continent* texts. Their average item difficulty in the real scenario obtained the best results (the nearest to 0.5) for all of the PoS. Moreover, in some cases, even the MCQs which were not supervised by the expert (Heuristic 6) obtained better results than those created manually by the expert, as it occurred with the verbs in the *Earth* (0.5 difficulty) and *Continent* texts (0.68). This shows the difficulty of the task and that automatic methods could be helpful to experts in the generation of items.

	Monosemous		Polysemous		NOT WN		Verb		Overall	
	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)	Diff. (%)	L > H (%)
Earth	83.33	66.67	96.67	60.00	83.33	83.33	100	88.89	92.98	68.42
Contin.	88.89	66.67	69.70	57.58	-	-	77.78	44.44	74.51	56.86
Bats	88.89	66.67	55.56	33.33	100	66.67	66.67	66.67	68.89	46.67
Arctic	73.33	60.00	66.67	53.33	33.33	100	44.44	33.33	61.90	54.76

Table VI.27: Analysis of the manually generated distractors

Table VI.27 shows the percentage of selected distractors together with the percentage of positive discrimination. With regard to the percentage of the number of the various selected distractors, the manual method obtained better results than the automatic methods in the *Earth* and *Continent* texts.

However, it tends to discriminate negatively more than all of the automatic methods, including the real scenario, although this negative discrimination is shown by a difference of at least two students for only three distractors: the distractor pilatzea (to accumulate) for the key bahitzea (to abduct);[16] the distractor babes (protection) for the key itzala (shadow);[17] and the distractor gaitzik (damage) for the key arazo (problem).[18] It is also work noting that the percentage of items selected by just one or two students is lower in comparison to the automatic methods.

In conclusion, even if the overall results of the expert are better than the results of the system, we cannot deny the fact that our research challenge is a difficult task: even the expert was unable to generate distractors of the same quality for different texts.

VI.4.3.4 Replacement of the keys with synonyms

Looking at the results for the manually generated distractors, there is a considerable difference (on average) in terms of item difficulty. Three of the texts obtained an average between 0.6 and 0.7, and the last one attained a value

[16] five students from the high-scoring group selected the distractors compared to 0 students from the low-scoring one.

[17] eight students from the high-scoring group selected the distractors compared to four students from the low-scoring one.

[18] nine students from the high-scoring group selected the distractors compared to three students from the low-scoring one.

of 0.46. As the human generator remained the same for all of the texts, we looked for reasons which were intrinsic to the texts. We found that, in the case of the *Continent* text, the number of different occurrences of some keys was significantly higher than in the other texts. Indeed, 64% of the keys appear in the aforementioned text more than once, with an average frequency of 7.81. As this text had the highest number of repetitions, we decided to study whether those repetitions could influence the results regarding difficulty. This was the reason for designing a new experiment, in which the keys of the items were replaced with a synonym in case there was more than one occurrence of such keys in the text.

Example VI.4.4 show a sample of the test in which the *keys* esperimentu (experiment), berotu (to warm) and dentsitatea (density) were replaced by their corresponding synonyms saiakuntza, epeldu and trinkotasun respectively. These three keys are a polysemous noun, a verb and a monosemous noun respectively.

Example VI.4.4 (Sample of the *Continent* text)
...2... erraz baten bidez egin zuten: hiru litroko depositu bat urez eta glizerinaz bete zuten, eta azpian xafla bero bat jarrita ...3... zuten likidoa. Ura berotzearekin batera, konbekzio-korronte bat eratu zen: azpialdean berotutako likidoak gora egiten zuen dentsitatea galdutakoan, eta gaineko likido hotzagoak beherantz, ...4... handiagoa zuelako. Hala, korronte zirkular. bat eratu zen.

(They carried it out by means of an easy ...2...: they first filled a three-litre deposit with water and glycerine and then they ...3... the liquid, placing a plate under the recipient. When the water warmed up, a convection current was formed: when the liquid warmed in the bottom and lost its density, it went up as the colder liquid came down, because it had a higher ...4..... In this way, a circular current was formed.)

We prepared two new tests: one using the manually generated distractors and the other using the distractors produced by Heuristic 6. In order to construct the two tests, we manually replaced the keys which appeared more than once with a synonym which did not appear in the text.

Table VI.28 shows the item difficulty indices of the MCQs that appear more than once in the text. With regard to the manually generated distractors, replacing the key with a synonym increased the difficulty value of seven out of eight items. In the case of the automatically generated distractors,

	Manually generated		Automatically generated	
	Original	Synonyms	Original	Synonyms
Item 2	0.66	0.16	0.84	0.48
Item 3	0.84	0.55	0.94	0.74
Item 4	0.84	0.39	0.75	0.56
Item 5	0.81	0.16	0.56	0.44
Item 7	0.81	0.39	0.84	0.52
Item 8	0.72	0.26	0.78	0.37
Item 16	0.59	0.42	0.81	0.89
Item 17	0.75	0.87	0.44	0.67

Table VI.28: Replacement using synonym: Comparison of item difficulty index

six out of the eight items were more difficult. Moreover, the difficulty of all of these items increases considerably. On the contrary, items which did not have a synonym as the key varied in terms of difficulty by just 0.1 points. Overall, the average item difficulty improved in both cases, from 0.66 to 0.48 in the case of the manually generated test and from 0.74 to 0.65 in the case of the automatically generated test.

The results confirm that different occurrences of the key within the text help students to complete the tests. Thus, our system should consider the option of replacing a key with a synonym if it appears more than once in the text.

VI.4.4 To sum up

We have defined six heuristics which depend on the PoS and polysemy of nouns. The results of the quantitative evaluation show that the best heuristic is to apply LSA plus ontological and morphological features for verbs and nouns that do not appear in WordNet, graph-based methods for monosemous nouns that appear in WordNet and a combination of the corpus-based and graph-based approaches for polysemous nouns that appear in WordNet.

In our opinion, there are three reasons to claim that this heuristic is the best one: first, it tends to get the best results in terms of average item difficulty and also generates the lowest number of easy items; second, compared to the other heuristics, students select the highest number of distractors in three out of four texts; and third, it achieved average results as regards negative discrimination among distractors.

When a qualitative evaluation was added to the generation process, the results confirmed that the obtained distractors were of better quality in comparison to the tests without any expert supervision.

Obviously, in general, the tests generated by the expert obtained better results and more difficult items were created. In addition, the distractors generated by the expert attracted more students than the automatically generated ones (see Tables VI.17 and VI.27). However, among the selected distractors, those generated by the expert tended to discriminate negatively more than the automatic methods. In addition, the system sometimes provides experts with new distractors that they would otherwise never produce.

In addition, the results prove that the items generated in a real scenario were similar to manually generated items. In fact, there are cases in which the real scenario obtained better results. In conclusion, these results confirm the appropriateness of such a system as a helpful tool for teachers. In addition, this scenario provides encouraging results for integrating our system into a real application.

In addition to comparing the manually and automatically generated items, the divergent results obtained in the manually generated tests suggest that the selected articles could have influenced the test. In fact, the experiment that involved replacing the keys with synonyms confirmed that having several occurrences of the key within the text helps students when they are completing the tests.

All in all, studies on generating science tests using semantic similarity measures constitute a promising research line for the future.

CHAPTER VII

Conclusions and future work

As we pointed out in the introductory chapter, ICTs are widely used in different scenarios as media and methodologies. In this dissertation, we have presented ICT as an approach to help in the learning process of certain subjects. Indeed, various institutions expend time and effort on the production of didactic resources and content. It is undeniable that the effort put into the creation of such resources leads to great results as regards their pedagogic appropriateness. In contrast, a large amount of this data is static, and after a certain period of time, could become outdated.

The analysis of various available NLP tools and corpora has demonstrated that it is possible to implement a system that helps experts and teachers in the creation of didactic material. Thus, we have designed and implemented a system called **ArikIturri** that, based on NLP and corpora, is able to produce items of a certain standard. ArikIturri is a multilingual system, and different question types have been tested in several scenarios. The representation of the items as well as the information relating to their generation process is carried out by means of a question model. This structured representation allows the importation and exportation of the items into independent applications.

We have conducted various experiments in which distinct linguistic information has been utilised. In our experiments, the input for the system is always a corpus, from which sentences are selected to be part of the items based on diverse criteria. In addition, their grammatical and semantic information enabled us to carry out experiments: (i) to prove the viability of the system designed to implement a complete automatic process to generate items; (ii) to apply different methods in the generation of distractors; and (iii) to modify some components of the source sentences when creating the stems. The results of these experiments were obtained from experts' opinions and students' answers. In this way, a qualitative analysis based on experts knowledge gave us a way of measuring the correctness of the automatically generated questions. In addition, the quantitative analysis based on students' responses ensured the quality of the items.

VII.1 Contributions

The contributions of the investigative work which were outlined in chapter I are described in more detail in the following.

ArikIturri

As previously mentioned, **the main contribution of this dissertation is the system called ArikIturri** (Aldabe *et al.*, 2006), the output items of which aim to form part of a test. As the creation of tests is a difficult task even for human editors, a system which produces a draft of a test can be seen as a helpful tool for experts and teachers (Coniam, 1997). Thus, we aim to facilitate their work. In fact, the automatic generation of items offers teachers (among others) the opportunity to use real-world texts in the form of sentences extracted from corpora as well as the option of selecting distractors that they would not produce manually.

The generation of items is based on different NLP tools and corpora. Before any generation approach was defined, the role that both resources can play in the generation process was studied. As a consequence, we have detected the impossibility of producing items designed to deal with some topics that are part of the Basque language learning curricula, due to the available corpus (Aldabe *et al.*, 2006). As regards the NLP tools, although the performance of the integrated tools is not perfect, their failures do not critically affect the performance of ArikIturri. Consequently, some minor errors by the verb conjugation and morphological declension tools have been observed and treated automatically by the system.

Based on the collected corpora, we have designed various scenarios in which various approaches have been tested. More specifically, we have proven the viability of the system to work in the Basque language learning, English language learning and science domains. In addition, the experiments have corroborated the feasibility of ArikIturri to produce several types of question: error correction, FBQs, word formation, MCQs and short answer questions. Therefore, **the applicability of defining a modular and multilingual system has been confirmed** by implementing methods designed to create different question types and evaluating the items in general and domain-specific scenarios.

In addition to ensuring the appropriateness of the system, it is also necessary to guarantee the accessibility of the automatically generated items. As we have mentioned, the inclusion of standards in our domain allows the option of offering sharable, reusable and accessible content. With this purpose, **we first defined and implemented a question model** with the aim of being as flexible and general as possible in order to represent different types of question (Aldabe *et al.*, 2007b). We considered it necessary to define our own model because the model includes information relating to the generation process. However, as we were aware of the existence of the QTI standard (IMS Global Learning Consortium, accessed 2010) that represents the test data and their corresponding results, we also worked on an extension point of QTI.

Evaluation

We have devoted our attention not only to the automatic generation of items, but also to their evaluation. In fact, **the evaluation of items is one of the strong contributions of this thesis**, and has sustained the capacity of the system to create useful items in order to test students' knowledge. Although a manual evaluation was used to test the generated items, we also considered an automatic evaluation as an instrument to detect problematic items. The *ill-formed question rejecter* module was integrated into the architecture of ArikIturri with the purpose of automatically evaluating items (Aldabe *et al.*, 2006). Thus far, we have noted the option of automatically discarding incomplete and badly-formed items. In addition, we have also focused on two features of the generated items, once the system produces an output. On the one hand, we analysed the correctness of the items based on experts' results. On the other hand, we measured the quality of the generated items by conducting several experiments with students. In this way, the manual evaluation was carried out qualitatively and quantitatively.

Using grammar when generating test items

The first steps towards the development of a completely automatic system involved investigating the use of linguistic information, and more specifically, grammatical information in various internal steps of the generation process. Therefore, the experiments based on the use of this information tested the correct performance of distinct modules of ArikIturri separately.

The *topic identification* and *sentence selection* tasks were conducted based on morphological information. All of the experiments which focused on the Basque language learning scenario took this information into account. In the case of the experiments which concentrated on English language learning, the system also needed some morphological information (the lemma and category of the keys) in order to find and select the topic and candidate sentences. Finally, in the science domain, the topic identification task and consequently the sentence selection task were performed manually.

The *item generator* module studied the techniques to construct interrogative stems from affirmative statements based on morphological and syntactic information (Aldabe *et al.*, 2011). First, the identification of the topic was based on NuERCB. Then, the *item generator* module carried out the required transformation and modification steps in order to obtain the indispensable components of the interrogative stems. In this way, the module also considered the morphological and syntactic information when carrying out this action. What is more, the identification and the generation of the corresponding wh-word was also conducted based on linguistic patterns. The evaluation results confirm the notable success of the automatic generation of grammatically correct questions.

The last component of the items which were generated taking into account grammatical information is the distractor element. The *distractor generator* module consulted grammatical information when the tests were focused on evaluating the grammar of Basque determiners, nouns and verbs in the Basque language learning scenario. The heuristics used in these experiments were established taking into account: (a) manually defined criteria; (b) errors detected in learner corpora; and (c) automatically extracted grammatical information.

All of the experiments which focused on the use of grammatical information were evaluated qualitatively. The evaluations focused on the correctness of the generated items and the results were obtained from experts' comments. Experts' opinions confirm the high degree of achievement in the automatic generation of correct items.

Using semantics when generating test items

Finally, we devoted time to investigating the use of semantic resources. The system exploits such information within the *distractor generator* module in order to create distractors which are semantically similar to the key of MCQs.

We designed two scenarios in order to test students' knowledge: English language learning and scientific vocabulary learning. Both scenarios were planned to test the knowledge of students and the applied methods were based on measures of relatedness in order to examine the vocabulary acquired by students. For this purpose, various corpus-based and graph-based approaches were studied. Based on the availability of the resources, distributional similarity measures were set as the starting point of the distractor generation task.

In the English language learning scenario, the tests were restricted to English verbs which appear in the AWL. These items were presented in isolation and the distractors were obtained based on the information radius measure (Aldabe *et al.*, 2009). On the one hand, the experiment which was performed proved the multilinguality of the system. On the other hand, English teachers who actually make use of this type of test in their classrooms analysed the automatically generated items. In this way, the correctness of the MCQs was estimated with experts who were familiar with the AWL, thereby making the evaluation more realistic. The results are promising as regards selecting different distractors taking into account the context of the candidate sentences.

Finally, the science scenario was conceived as a real scenario in which the entire testing process was simulated. With this purpose, the Basque noun and verb candidate distractors were extracted from an LSA model (Aldabe and Maritxalar, 2010), and several approaches were checked. The items were verified as part of a entire text, due to the option of receiving help from experts and teachers who conduct this type of test in class. In total, 18 schools and 951 second-grade OSE students took part in the experiments. The results confirm the appropriateness of offering such a system as a helpful tool for the generation of science tests.

VII.2 Conclusions

The conclusions presented below have been derived from the contributions presented in the previous section and from the observed results relating to them.

ArikIturri

The methodology which was applied in order to build ArikIturri has proven the viability of the system to establish different criteria in order to generate several question types and topics. Therefore, the experiments have confirmed the appropriateness of designing a modular and multilingual system. Thus, we have built a complete and complex system.

The main constraint of our system is the use of computationally expensive NLP tools. For instance, the acquisition of a high number of items in a row can take time due to the process of analysing the texts at the chunk level. In cases in which this linguistic information is not indispensable, it could be possible to avoid this analysis. Furthermore, ArikIturri could use a simpler analyser to perform a real time execution. Otherwise, a Web application should require a two-step process to make the request more dynamic for the final user.

Question model

The decision to first define a question model and then propose an extension point of QTI may be reprehensible to researchers who consider it indispensable to provide information in a standardised way. Nonetheless, it was more important for us to be able to represent information relating to the generation process, so that we prioritised building our own model. As a consequence, the items generated by ArikIturri can be represented in two different ways. Thus, based on the environment responsible for requesting tests from ArikIturri, the output is represented either by the question model or by QTI. In addition, our system also provides an alternative to transforming the output represented by means of our question model into a QTI representation and vice versa. It is clear that both models have been implemented with one purpose.

Our question model supplies a complete representation of the linguistic features of items. This representation allows, among other things: (i) the experts' understanding of the generation process; (ii) the improvement of the system by means of the feedback provided by experts; and (iii) the explicit representation of linguistic features in authoring tools or post-editing environments.

In contrast, the QTI representation increases the accessibility of the content generated by ArikIturri. Applications that make use of the QTI representation are usually more focused on the exchanging of data or the storing of student responses. In fact, this is the main purpose of QTI. As a consequence, when an application requests ArikIturri to generate items to be presented by QTI, losing some linguistic information is not critical.

Experts' evaluation

We have developed a post-editing environment that allowed us to store the experts' evaluation results. One positive additional consequence of making use of the post-editing environment is related to the claimed independence of ArikIturri from any application. Employing an external and real application to evaluate the items has demonstrated the process of importing the items generated by ArikIturri.

The items imported into the post-editing environment provided us with contributions from experts. On the one hand, the feedback obtained from human evaluators facilitated the improvement of the post-editing environment. On the other hand, their comments as regards the generation process supplied us with enhancements.

The manual evaluation and the collection of experts' opinions have been an invaluable source with regard to corroborating some hypotheses which were defined when developing the system. First of all, the experiments which focused on the correctness of the items have proven that the automatic generation of knowledge construction reduces the time spent by teachers on constructing exercises (Aldabe *et al.*, 2006). Second, the experts' opinions have given us hints as to how to improve the automatic process. For instance, the experts proposed changing the position of the blank if it was at the beginning of the question when the topic was related to Basque verb forms (Aldabe *et al.*, 2007a). Third, the manual construction of complete MCQs provided us with new criteria with which to generate items. Finally, the comparison of automatically generated distractors with those which were produced manually (Aldabe and Maritxalar, 2010) demonstrated that the system is able to produce some valid distractors that experts would not create manually.

Language learning scenarios

In the Basque language learning scenario, ArikIturri generated items to deal with the correct use of determiners, nouns and verbs. These experiments considered three strategies with which to define heuristics. The results derived from the manually defined heuristics confirm that it is an interesting approach, but this method involves a laborious methodology. Thus, in a second attempt, the system based the distractor generation task on errors detected in learner corpora. Finally, in an effort to automatise the entire definition process, we studied automatically extracted patterns. These latter two strategies can provide a good resource with which to define heuristics and can imply a more practical methodology.

The English language learning scenario was designed to generate items relating to the vocabulary of verbs. The purpose of this set of experiments was twofold. On the one hand, we verified the multilinguality of ArikIturri. On the other hand, we started to deal with semantics within the distractor generation task.

In conclusion, although the heuristics based on grammatical information are a good attempt to generate items automatically, the creation of distractors which are similar to the key when dealing with vocabulary is a more complex course of action, and so a more interesting field to investigate.

Science domain tests

Based on the previous premise, and due to the opportunity to evaluate some tests in a real scenario, this dissertation devoted special attention to the study of various corpus-based and graph-based semantic measures for defining heuristics. All in all, six heuristics were established and tested with students in the science domain.

The results show that there is no particular heuristic that significantly outperforms the rest. In general, the heuristic which applies different approaches based on the PoS and semantic features of the key attained the best results. In the case of verbs and nouns that do not appear in WordNet, the method which is based on the LSA model and takes into account ontology and morphological features is the best strategy. For monosemous nouns that appear in WordNet, the graph-based method is the best one. Finally, for polysemous nouns, the method that combines the LSA and graph-based approaches is the most promising approach. There are three reasons to claim that this is the best heuristic. First of all, it tends to get the best results in terms of item difficulty and also generates the lowest number of easy items. Second, the distractors generated by this heuristic attract more students in general. Finally, it achieved average results as regards the creation of distractors that needed to be revised.

In a real scenario, the item analysis of the automatically generated items should be a two-step process: a quantitative analysis and then a qualitative analysis. The results confirm that these supervised tests are more favourable than those tests without any supervision.

In one last experiment, we went one step further in order to measure the quality of the heuristics. We asked an expert to manually generate the distractors in order to compare them with the automatically generated tests. This is why the expert had to base the creation on semantic measures in the science domain. At this point, it is necessary to note the high degree of expertise of the human generator who took part in this process. Obviously, in general, the items generated by the expert obtained better results in terms of difficulty, and more demanding items were created. However, the distractors selected by the students in the examination task tended to feature more negative discrimination than the items which were generated automatically. Finally, in some cases, we detected that ArikIturri provides experts with some distractors that they would not otherwise produce. In addition, the results also prove that the items generated in a real scenario were similar to manually generated items. In fact, there were cases in which the real scenario obtained better results.

All in all, we can conclude that the investigation of generating science tests using distributional similarity and graph-based measures is a promising research line. In fact, some teachers who took part in the in-class experiments expressed great interest in employing ArikIturri as a tool.

Question generation challenge

In addition to focusing on various topics and several scenarios, we worked with the transformation of declarative statements into interrogative stems. We created questions (interrogative statements) in order to ask about numerical entities. Based on these entities, the WHICH, HOW MANY and WHEN wh-words were identified. This approach proved the viability of generating grammatically correct questions in a completely automatic way.

This inclusion in the QG challenge has opened up a new research line in the IXA research group. Although our research group covers a wide variety of research areas, including the question answering task, no work had previously been proposed as regards QG and the Basque language.

The challenge itself is fairly new in the research community, as it was first proposed in 2008 (Nielsen, 2008). Thus, our proposal aims to be part of this new challenge in which people from diverse backgrounds are taking part and working on different scenarios. As a novelty to offer to the community, to our knowledge, our proposal is the first to deal not only with English, and to use a multilingual perspective.

VII.3 Future work

This last section presents some open research lines as well as new research lines that should be considered in order to improve the system in the future.

Topic selection

The tests generated in the science domain were aimed at working with the vocabulary of a given text. For this purpose, the meaningful terms were manually marked. Nonetheless, once the appropriateness of generating distractors based on semantic information was corroborated, it became necessary to automatise the process of detecting the meaningful terms in the text. As previously mentioned, we plan to incorporate a term extractor for Basque, such as Erauzterm (Alegria *et al.*, 2004). However, as Erauzterm extracts the terminology of an entire corpus, before integrating it into ArikIturri, an adaptation of the extractor will be necessary in order to find the methodology to obtain appropriate terms from single documents. In the same way, we contemplated analysing data mining techniques in order to apply them to the identification of blanks.

Scenarios, evaluation and assessment

The results regarding vocabulary in the science domain have proven the applicability of the use of semantic resources in the generation of distractors. New scenarios should be considered in order to improve the task and to define new strategies.

We have already mentioned that the expert who took part in these experiments had a high level of expertise in the task of creating didactic resources. As a consequence, the tests the expert generated have to be considered as an upper bound. Furthermore, the help offered by ArikIturri to this type of users may sometimes be limited. In contrast, teachers are usually less experienced as a community in the creation of content. As a consequence, we believe that ArikIturri would be of more help to them in the generation task in comparison to experts. In order to corroborate this assumption, we are planning to carry out new experiments with teachers from schools and to leave aside the experts.

As previously specified, we obtained a collection of results from 18 schools from all over the Basque Country. More specifically, we tested the items with students whose mother tongue is Basque, but their dialect varied depending on the area they came from. For this reason, future research should also take into account the influence of the sociological characteristics of the students.

All of our research has focused on the evaluation of the system. However, the final goal of ArikIturri should be to produce items that assess students with regard to some particular competences. In accordance with this purpose, reliability and validity measures should be considered in future research.

Question generation challenge

The QG challenge has established a group of multidisciplinary researchers whose two main concerns are the automatic generation of questions and the generation of relevant questions. Thus, while the former approach exploits a wide variety of NLP tools and linguistic resources, the second pays more attention to the pedagogical importance of the questions.

In our work, we have focused on the challenge of generating the questions automatically. In addition to the previously implemented wh-words, we are planning to add new ones as part of this research line. We plan to generate WHO, WHOM and WHERE questions based on the entities classified by the Named Entity Recogniser for Basque (Alegria *et al.*, 2003b). In addition, we plan to incorporate semantic information into the stem generation task. With this purpose, we intend to use the semantic role labelling for Basque (Aldezabal *et al.*, 2010) to deal with wh-words.

Undoubtedly, we are also aware of the importance of generating questions that test the essential concepts of a given text. In our case, we are particularly interested in reading comprehension tasks, for which we plan to create a computer-assisted assessment scenario. The purpose of this is to define an environment in which, given an input text, the system will generate MCQs to test students' comprehension. Thus, each MCQ will contain an interrogative stem which will enquire about relevant concepts of the text. In addition, in this type of items, not only the distractors but also the correct answer should be generated automatically. The research line started by Chen *et al.* (2009) has pointed out the usefulness of applying a situation model in order to generate questions for the reading strategy of self-questioning. Accordingly, we also intend to build a model of concepts extracted from the input text and then, based on this model, to generate MCQs designed to test the knowledge of students.

Applications

Finally, we plan to implement a Web application in which teachers would upload a text and mark some terms. After that, our system would propose some candidate distractors, thereby helping the teachers to produce their own tests. The idea of developing this particular instantiation of ArikIturri came from the opinions which were collected when conducting in-class experiments.

Bibliography

ADL. *SCORM 2004 4th Edition*, 2009. URL `http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th/`.

Aduriz I., Agirre E., Aldezabal I., Alegria I., Ansa O., Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza, Ezeiza N., Gojenola K., Maritxalar M., Oronoz M., Sarasola K., Soroa A., Urizar R. and Urkia M. *A framework for the automatic processing of Basque*. In *Proceedings of Workshop on Lexical Resources for Minority Languages*. Granada, 1998.

Aduriz I., Aranzabe M., Arriola J., Atutxa A., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Oronoz M., Soroa A. and Urizar R. *Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing*. Corpus Linguistics Around the World. Book series: Language and Computers, volume 56, pp. 1–15, 2006.

Aduriz I., Aranzabe M., Arriola J., Díaz de Ilarraza A., Gojenola K., Oronoz M. and Uria L. *A cascaded syntactic analyser for Basque*. Computational Linguistics and Intelligent Text Processing, pp. 124–135, 2004.

Agirre E., Ansa O., Arregi X., Artola X., Díaz de Ilarraza A., Lersundi M., Martinez D., Sarasola K. and Urizar R. *Extraction of semantic relations from a Basque monolingual dictionary using constraint grammar*. In *Proceedings of Euralex Sttutgart (Germany). 2000. ISBN 3-00-006574-1*, 2000.

Agirre E. and Martinez D. *Integrating selectional preferences in WordNet*. In *Proceedings of first international WordNet conference*. Mysore, India, 2002.

Agirre E. and Soroa A. *Personalizing pagerank for word sense disambiguation*. In *Proceedings of EACL-09*, pp. 33–41. Athens, Greece, 2009.

Aldabe I., Lopez de Lacalle M. and Maritxalar M. *Automatic acquisition of didactic resources: generating test-based questions*. Proceeding of SINTICE 07 (Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación), pp. 105–111, 2007a.

Aldabe I., Lopez de Lacalle M., Maritxalar M. and Martinez E. *The question model inside ArikIturri*. In *Proceedings of the 7th IEEE International Conference on Advance Learning Technologies (ICALT 2007)*, pp. 758–759. Niigata, Japan, 2007b.

Aldabe I., Lopez de Lacalle M., Maritxalar M., Martinez E. and Uria L. *Arikiturri: An automatic question generator based on corpora and NLP techniques*. In *Proceedings of the Eigth International Conference on Intelligent Tutoring Systems (ITS'06)*, pp. 584–594. Jhongli, Taiwan, 2006.

Aldabe I. and Maritxalar M. *Automatic distractor generation for domain specific texts*. In *Proceedings of the 7th International Conference on NLP, IceTAL 2010*, pp. 27–38, 2010.

Aldabe I., Maritxalar M. and Mitkov R. *A study on the automatic selection of candidate sentences and distractors*. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009)*, pp. 656–658. Brighton, UK, 2009.

Aldabe I., Maritxalar M. and Soraluze A. *Question generation based on numerical entities in Basque*. In *Proceedings of AAAI Symposium on Question Generation*, p. submitted, 2011.

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Oronoz M. and Sarasola K. *Application of finite-state transducers to the acquisition of verb subcategorization information*. Natural Language Engineering. Cambridge University Press., volume 9(1), pp. 39–48, 2003.

Aldezabal I., Aranzabe M., Díaz de Ilarraza A., Estarrona A. and Uria L. *EusPropBank: Integrating semantic information in the Basque dependency treebank*. In A. Gelbukh, ed., *Computational Linguistics and Intelligent Text Processing*, volume 6008/2010 of *Lecture Notes in Computer Science*, pp. 60–73. Springer Berlin / Heidelberg, 2010.

Alegria I., Aranzabe M., Ezeiza A., Ezeiza N. and Urizar R. *Robustez y flexibilidad de un lematizador/etiquetador*. In *VIII Simposio Internacional de Comunicacion Social*. Santiago de Cuba, 2003a.

Alegria I., Arrieta B., Carreras X., Díaz de Ilarraza A. and Uria L. *Chunk and clause identification for Basque by filtering and ranking with perceptrons*. In *Proceedings of SEPLN*. Madrid. Spain, 2008.

Alegria I., Etxeberria I., Hulden M. and Maritxalar M. *Porting Basque morphological grammars to foma, an open-source tool*. Finite-State Methods and Natural Language Processing Lecture Notes in Computer Science, volume 6062/2010, pp. 105–113, 2010.

Alegria I., Ezeiza N., Fernandez I. and Urizar R. *Named entity recognition and classification for texts in Basque*. In *II Jornadas de Tratamiento y Recuperación de Información, JOTRI, Madrid. 2003. ISBN 84-89315-33-7*, 2003b.

Alegria I., Gurrutxaga A., Lizaso P., Saralegi X., Ugartetxea S. and Urizar R. *Linguistic and statistical approaches to Basque term extraction*. In *GLAT-2004: The Production of Specialized Texts. ISBN: 2-908849-14-3*, 2004.

Aranzabe M. *Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitz-bankua eta gramatika konputazionala*. Ph.D. thesis, Euskal Filologia Saila (UPV/EHU), 2008.

Areta N., Gurrutxaga A., Leturia I., Alegria I., Artola X., Díaz de Ilarraza A., Ezeiza N. and Sologaistoa A. *ZT corpus: Annotation and tools for Basque corpora*. In *Copus Linguistics*. Birmingham, UK, 2007.

Artola X., Díaz de Ilarraza A., Soroa A. and Sologaistoa A. *Dealing with complex linguistic annotations within a language processing framework*. IEEE Transactions on Audio, Speech, and Language Processing. Vol 17, number 5. Pages 904-915. ISSN: 1558-7916, 2009.

Atserias J., Villarejo L., Rigau G., Agirre E., Carroll J., Magnini B. and Vossen P. *The meaning multilingual central repository*. In Proceedings of the Second International WordNet Conference-GWC, pp. 23–30, 2004.

Becker L., Nielsen R., Okoye I., Sumner T. and Ward W. *Whats next? target concept identification and sequencing*. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pp. 35–44, 2010.

Bloom B.S. *Taxonomy of educational objectives*. Handbook I: The Cognitive Domain, 1956.

BNC Consortium. *The British National corpus, version 3 (BNC XML Edition)*. http://www.natcorp.ox.ac.uk, 2007. Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

Boyer K. and Piwek P., eds. *Proceedings of QG2010: The Third Workshop on Question Generation*. Pittsburgh: questiongeneration.org, 2010.

Boyle A., Russell T., Smith S. and Varga-Atkins T. *The elearning place: progress report on a complete system for learning and assessment*. In *8th International CAA Conference*, pp. 71–77, 2004.

Brants T. and Franz A. *Web 1t 5-gram corpus version 1*. Linguistic Data Consortium, 2006.

Brin S. and Page L. *The anatomy of a large-scale hypertextual web search engine*. Computer Networks and ISDN Systems, volume 30(1-7), 1998.

Carreras X., Màrquez L. and Castro J. *Filtering–ranking perceptron learning for partial parsing*. Machine Learning, volume 60, pp. 41–71, 2005.

CEFR, 2001. *Common European Framework of Reference for Languages*. Cambridge University Press, 2001. ISBN 9780521005319.

Chen C., Liou H. and Chang J. *FAST - An Automatic Generation System for Grammar Tests*. Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pp. 1–4, 2006.

Chen W., Aist G. and Mostow J. *Generating questions automatically from informational text*. In *2nd Workshop on Question Generation*, pp. 17–24. Brighton,UK, 2009.

Cohen J. *A coefficient of agreement for nominal scales*. Educational and Psychological Measurement, pp. 37–46, 1960.

Conejo R., Guzmán E., Millán E., Trella M., Pérez-De-La-Cruz J. and Ríos A. *SIETTE: A web-based tool for adaptive testing*. In *International Journal of Artificial Intelligence in Education*, volume 14(1), pp. 29–61, 2004.

Coniam D. *A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests*. CALICO Journal, volume 16(2–4), pp. 15–33, 1997.

Coxhead A. *A new academic word list*. In W. Teubert and R. Krishnamurthy, eds., *Corpus Linguistics: Critical Concepts in Linguistic*, pp. 123–149. Oxford:Routledge, 2000.

Dagan I., Lee L. and Pereira F. *Similarity-based methods for word sense disambiguation*. In *Proceedings of the Association for Computational Linguistics*, pp. 56–63, 1997.

Deerwester S., Dumais S., Furnas G., Landauer T. and Harshman R. *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, volume 41(6), pp. 391–407, 1990.

Díaz de Ilarraza A., Mayor A. and Sarasola K. *Semiautomatic labelling of semantic features*. Proceedings of the 19th International Conference on Computational Linguistics, 2002.

Dorow B. and Widdows D. *Discovering corpus-specific word senses*. In Proceedings of EACL, 2003.

Elhuyar Hizkuntza Zerbitzuak, ed. *Elhuyar Zientzia eta Teknologiaren Hiztegi Entziklopedikoa*. Elhuyar Edizioak/Euskal Herriko Unibertsitatea, 2009.

Ezeiza N. *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile sintaktiko sendo eta malgua*. Ph.D. thesis, University of the Basque Country, 2002.

Fellbaum C., ed. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

Fleiss J. *Measuring nominal scale agreement among many raters*. Psychological Bulletin, volume 76(5), pp. 378–382, 1971.

Gruber T. *Ontology*. In L. LIU and M.T. ZSU, eds., *Encyclopedia of Database Systems*, pp. 1963–1965. Springer US, 2009. ISBN 978-0-387-39940-9. 10.1007/978-0-387-39940-9_1318.

Guzmán E., Machuca E., Conejo R. and Libbrecht P. *LeActiveMath Integrated Adaptative Assessment Tool*. LeActiveMath: EC Sixth Framework Programme for Research and Technological Development, 2005. Deliverable D16.

HABE. *Helduen Euskalduntzearen Oinarrizko Kurrikulua*. HABE, Donostia, 1999.

Haveliwala T.H. *Topic-sensitive pagerank*. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pp. 517–526. ACM, New York, NY, USA, 2002. ISBN 1-58113-449-5.

Heilman M. and Smith N. *Extracting simplified statements for factual question generation*. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pp. 11–20, 2010.

Hoshino A. and Nakagawa H. *Assisting cloze test making with a web application*. Proceedings of SITE (Society for Information Technology and Teacher Eduation), pp. 2807–2814, 2007.

Hulden M. *Foma: a finite-state compiler and library*. In *Proceedings of EACL 2009*, pp. 29–32, 2009.

IEEE. *IEEE Standard for Learning Object Metadata*, 2002. URL `http://ltsc.ieee.org/wg12/`.

IMS Global Learning Consortium. *IMS Question & Test Interoperability Specification*, accessed 2010. URL `http://www.imsglobal.org/question/`.

Kerejeta M., Larrañaga M., Rueda U., Arruarte A. and Elorriaga J. *A computer assisted assessment tool integrated in a real use context*. In *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies. ICALT*, pp. 848–852, 2005.

Kilgarriff A. *Putting frequencies in the dictionary*. International Jounral of Lexicography, volume 10(2), pp. 135–155, 1997.

Kilgarriff A., Husak M., McAdam K., Rundell M. and Rychly P. *GDEX: Automatically finding good dictionary examples in a corpus*. Proceedings of EURALEX, 2008.

Kneser R. and Ney H. *Improved backing-off for m-gram language modeling*. In *International Conference on Acoustics, Speech, and Signal Processing*, pp. 181–184, 1995.

Koskenniemi K. *Two-level Morphology: A general Computational Model for Word-Form Recognition and Production*, 1983.

Laka I. *A Brief Grammar of Euskera - The Basque Language*. University of the Basque Country, Office of the Vice-Rector for the Basque Language, 1996. ISBN 84-8373-850-3. URL `http://www.ehu.es/grammar/`.

Landauer T.K., McNamara D.S., Dennis S. and Kintsch W., eds. *Handbook of Latent Semantic Analysis*. Mahwah NJ: Lawrence Erlbaum Associates, 2007.

Landis J. and Koch G. *The measurement of observer agreement for categorical data*. Biometrics, volume 33(1), pp. 159–174, 1977.

Lee J. and Seneff S. *Automatic Generation of Cloze Items for Prepositions*. Proceedings of Interspeech, pp. 2173–2176, 2007.

Liu C.L., Wang C.H., Gao Z.M. and Huang S.M. *Applications of Lexical Information for Algorithmically Composing Multiple choice Cloze Items*. 2nd Workshop on Building Educational Applications Using NLP, 2005.

Manning C. and Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

Maritxalar M. *Mugarri: Bigarren Hizkuntzako ikasleen hizkuntza ezagutza eskuratzeko sistema anitzeko ingurunea*. Ph.D. thesis, Informatika Fakultatea, UPV-EHU, 1999.

Martinez E. *Arikiturri: Corpusen erabilera hizkuntzaren ikaskuntzan*. Master's thesis, University of the Basque Country, 2005.

Mavrikis M. *MathQTI and the Serving Mathematics project*, 2005. URL `http://www.mathstore.ac.uk/articles/maths-caa-series/jan2005/index.shtml`.

Mitkov R., Ha L. and Karamanis N. *A computer-aided environment for generating multiple-choice test items*. Natural Language Engineering. Cambridge University Press, volume 12(2), pp. 177–194, 2006.

Mitkov R., Ha L.A., Varga A. and Rello L. *Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation*. Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pp. 49–56, 2009.

Morris J. and Hirst G. *Non-classical lexical semantic relations*. In *Workshop on Computational Lexical Semantics, Human Language Technology Conference of the North American Chapter of the ACL*, pp. 46–51. Boston, MA, 2004.

Nielsen R. *Question generation: Proposed challenge tasks and their evaluation*. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*. NSF, Arlington, VA, 2008.

Nikolova I. *New issues and solutions in computer-aided design of MCTI and distractors selection for Bulgarian*. Proceedings of Multilingual Resources, Technologies and Evaluation for Central and Eastern European Languages, pp. 40–46, 2009.

Niles I. and Pease A. *Towards a standard upper ontology*. In *Proceedings of the 2nd international conference on formal ontology in information systems, FOIS 2001*. Ogunquit, Maine, 2001.

Oronoz M. *Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura*. Ph.D. thesis, Lengoaia eta Sistema Informatikoak Saila, Euskal Herriko Unibertsitatea, 2009.

Pino J., Heilman M. and Eskenazi M. *A Selection Strategy to Improve Cloze Question Quality*. Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems., 2008.

Pociello E., Agirre E. and Aldezabal I. *Methodology and construction of the Basque WordNet*. Language Resources and Evaluation. Springer Netherlands, 2010.

Resnik P. *Using information content to evaluate semantic similarity in a taxonomy*. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

Rus V. and Graesser A., eds. *The Question Generation Shared Task and Evaluation Challenge*, 2009. ISBN 978-0-615-27428-7.

Sarasola I. *Euskal Hiztegia*. Donostia, Gipuzkoako Kutxa, 1996.

Smith S., Kilgarriff A., Sommers S., Wen-liang G. and Guang-zhong W. *Automatic cloze generation for English proficiency testing*. Proceeding of LTTC conference, 2009.

Smith S., Sommers S. and Kilgarriff A. *Learning words right with the Sketch Engine and WebBootCat: Automatic cloze generation from corpora and the web*. Proceedings of CCU, 2008.

Soraluze A., Alegria I., Ansa O., Arregi O. and Arregi X. *Recognition and classification of numerical entities in Basque*. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*, 2011.

Sumita E., Sugaya F. and Yamamota S. *Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions*. 2nd Workshop on Building Educational Applications Using NLP, 2005.

Tapanainen P. and Jarvinen T. *A non-projective dependency parser*. Proceedings of the 5th Conference on Applied Natural Language Processing, pp. 64–71, 1997.

Turney P. *Mining the web for synonyms: PMI-IR versus LSA on TOEFL*. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pp. 491–502. Freiburg, Germany, 2001.

Uria L. *Euskarazko erroreen eta desbideratzeen analisirako lan-ingurunea. Determintzaile-erroreen azterketa eta prozesamendua*. Ph.D. thesis, Euskal Filologia Saila, Euskal Herriko Unibertsitatea, 2009.

Uria L., Arrieta B., Díaz de Ilarraza A., Maritxalar M. and Oronoz M. *Determiner errors in Basque: Analysis and automatic detection*. In *Proceedings de XXV Conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN)*, pp. 41–48, 2009.

Vale C.D. *Computerized item banking*. In S.M. Downing and T.M. Haladyna, eds., *Handbook of Test Development*, pp. 261–285. Lawrence Erlbaum Associates, 2006.

Vossen P., ed. *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*. Kluwer, Dordrecht, 1998.

West M. *A General Service List of English Words*. Longman, London, 1953.

Yao X. *Question Generation with Minimal Recursion Semantics*. Master's thesis, Saarland University & University of Groningen, 2010. URL `http://cs.jhu.edu/~xuchen/paper/Yao2010Master.pdf`.

Zesch T. and Gurevych I. *Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words*. Journal of Natural Language Engineering., volume 16(01), pp. 25–59, 2010.

Glossary

answer focus

The answer focus needs to be understood as the minimum amount of information required in order to deal with the topic. Thus, the answer focuses are the chunks of the sentence in which the relevant topic appears.

assessment

Assessment measures the performance of learners with regard to a set of competencies such a knowledge, skills and attitudes.

distractor

A distractor is an incorrect choice among multiple-choice answers on a test.

error correction

Error correction items consist of a sentence with at least one error that students have to correct. The error, which can be marked or unmarked, is a distractor which is generated automatically by the system.

evaluation

Evaluation is the process of determining the value of the items in order to accept, modify or reject them.

fill-in-the-blank question (FBQ)

Fill-in-the-blank items require students to complete a statement by supplying a brief response. In some cases, an FBQ can be a question which students have to answer with a brief response.

item analysis

Item analysis theory reviews items qualitatively and quantitatively, with the aim of identifying problematic items. The qualitative analysis is usually based on experts' knowledge, whereas the quantitative analysis is conducted after the items have been given to students, i.e., statistical analysis.

item bank

An item bank is more than a collection of items or questions, as the items usually have different properties which lead to the specification of the information relating to their administration and scoring.

item difficulty

The difficulty of an item can be described as the proportion of students who answer the item correctly. The higher the difficulty value, the easier the item.

item discrimination

This index indicates the discriminatory power of an item. That is, an item is effective if those with high scores tend to answer it correctly and those with low scores tend to answer it incorrectly.

item or question

The terms item and question are used to refer to the output of ArikIturri. Both terms are used interchangeably, even if not all items contain interrogative statements. The term item can be seen as a more general term. In fact, the term item covers a variety of concepts, as tests are not always collections of questions, but problems to solve or even assertions to evaluate.

Broadly speaking, a question or item is composed of a stem that requires an answer (key). The stem is the part of the item that presents the item as a problem to be solved, a question or an incomplete statement. In addition, depending on the type of question, an item can also be composed of a list of distractors, a distractor being an incorrect choice among multiple-choice answers on a test.

key

The key is the correct answer to the stem of the question.

multiple-choice question (MCQ)

Multiple-choice items consist of a stem and a set of options. The stem is the first part of the item and presents the item as a problem to be solved, a question or an incomplete statement. The options are the possible answers that the students can choose from, with the correct answer (the key) and the incorrect answers (distractors).

QTI

The IMS Question and Test Interoperability specification is a standard for representing questions and test data and their corresponding results, enabling the exchange of data across different LMSs.

reliability

Reliability is obtained when the same test is evaluated with the same group of students in different periods and the results obtained are the same.

short answer question

Short answer items require students to respond to a question by generating a brief text or response. We have distinguished two groups of short answer questions which are created by ArikIturri. Both comprise interrogative statements, but while there are some questions in which the system offers a clue to the answer as a help to students, there are others that consist of just the questions that the students have to answer.

stem

The stem is the part of the item that presents the item as a problem to be solved, a question or an incomplete statement. Thus, the stem can be a declarative or interrogative statement. It can also be an incomplete sentence (containing a blank), and the correct answer to the stem is the key of the question.

topic

The topic is the concept that students have to work with and is part of their curriculum. From an item banking point of view, this concept can be seen as the stimulus of the item. From a more pedagogical point of view, a topic is a concept that students work with during their learning process.

This term comprises a range of concepts, from the simplest unit of work to the most complex. For instance, the topic of an item could be the conjugation of a concrete verb or reading comprehension.

validity

Validity is computed in order to ensure that the test measures what it is intended to measure.

word formation

Word formation items consist of a sentence with a blank and a word the form of which must be changed in order to fit it into the gap.

APPENDIX A

Basque linguistic phenomena

Basque Morphology

Basque is an agglutinative language with a rich morphology. The most prominent features of the language include (Alegria *et al.*, 2010):

- Basque morphology is very rich. The determiner, the number and the declension case morphemes are appended to the last element of the noun phrase and always occur in this order;

- Basque nouns belong to a single declension; the 15 case markers are invariant;

- Functions normally fulfilled by prepositions are realised by case suffixes inside word-forms. Basque offers the possibility of generating a large number of inflected word-forms. From a single noun entry, a minimum of 135 inflected forms can be generated. While 77 of these are simple combinations of number, determiners, and case marking (and not capable of further inflection), the rest (58) include one of the two possible genitive markers (possessive and locative) to which new declension cases can be appended. Due to this, Basque is considered an agglutinative language;

- Basque has ergative case, which marks the subjects of transitive verbs. Linguistic theories and nomenclature about this phenomenon are vary-ing: some use the terminology “ergative language”, and others “non-accusative;”

- The verb provides all the grammatical and agreement information about the subject and the two possible objects, as well as tense and aspect-related information, etc.

Declension system

		Singular	**Plural**	**Indefinite**
Absolutive	NOR	-a	-ak	-
Ergative	NORK	-ak	-ek	-(e)k
Dative	NORI	-ari	-ei	-(r)i
Possesive genitive	NOREN	-aren	-en	-(r)en
Comitative	NOREKIN	-arekin	-ekin	-(r)ekin
Benefactive	NORENTZAT	-arentzat	-entzat	-(r)entzat
Motivative	NORENGATIK	-arengatik	-engatik	-(r)engatik
Inessive	NON	-(e)an	-etan	-(e)tan
Locative genitive	NONGO	-(e)ko	-etako	-(e)tako
Ablative	NONDIK	-(e)tik	-etatik	-(e)tatik
Allative	NORA	-(e)ra	-etara	-(e)tara
End-point allative	NORAINO	-(e)raino	-etaraino	-(e)taraino
Directional allative	NORANTZ	-(e)rantz	-etarantz	-(e)tarantz
Destinative allative	NORAKO	-(e)rako	-etarako	-(e)tarako
Instrumental	NORTAZ	-az	-etaz	-(e)z/-taz

Table A.1: 15 case makers

One of the principal characteristics of Basque is its declension system with numerous cases. The inflections of determination, number and case appear only after the last element in the noun phrase. This last element may be the noun, but also typically and adjective or a determiner. For example:
etxe zaharreAN (in the old house)
etxe: noun (house)
zahar: adjective (old)
r and e: epenthetical elements

A: determinate, singular
N: inessive case
The 15 case markers are invariant and are presented in Table A.1.

Verbal inflection

In general, a verb can have from one to four different auxiliary paradigms. These paradigms correspond to the following four auxiliary types:

- DA: One-argument intransitive verbs. The absolutive is the subject of the clause;
- DU: Two-argument transitive verbs. The ergative is the subject and the absolutive is the direct object of the clause;
- DIO: Three-argument transitive verbs. The ergative is the subject, the absolutive is the direct object and the dative is the indirect object of the clause;
- ZAIO: Two-argument intransitive verbs. The absolutive is the subject and the dative is the indirect object of the clause.

Determiner errors

From (Uria *et al.*, 2009):

> Basque is an agglutinative language in which most words are formed by joining morphemes together and it is said to be a free-word-order language because the order of the phrases in a sentence can vary. On the contrary, the order of the elements that constitute the noun phrase (NP) is fixed: nouns head the NPs, adjectives follow the nouns and determiners (articles and demonstratives) follow the [Noun + Adj] groups; other modifiers such as possessive phrases, postpositional phrases, relative clauses and most quantifiers always precede the nouns. From the point of view of generative linguistics, the determiner, in general, appears in the last position of the NP, in some cases agglutinated to a word, and it takes the entire NP as its complement, constituting the Determiner Phrase (DP) (Laka, 1996). The following examples show a few types of correct determiners and DP structures:

- the singular and plural definite articles: -a / -ak (the), which in Basque are suffixes to nouns and adjectives:

  [[[haurraren]$_{GEN}$ jostailu]$_{NP}$ -a/-ak]$_{DP}$
  child of toy(s) the

  (the toy(s) of the child)

- the singular and plural indefinite articles: bat (one) / batzuk (some, ones)

  [[[haurraren] jostailu]$_{NP}$ bat / batzuk]$_{DP}$
  child of toy(s) a/some

  (a / some toy(s) of the child)

- the demonstratives: hau / hori / hura (this/that) / hauek / horiek / haiek (these/those):

  [[[haurraren] jostailu]$_{NP}$ hau/hori/hura]$_{DP}$
  child of toy(s) this/that/these/those

  (this/ that / these / those toy(s) of the child)

However, depending on some characteristics of the DP, the use of determiners may vary. Below some correct and incorrect examples of Basque DPs are showen:

- Arguments require a determiner:

  emakumea etorri da (the woman has arrived)

  *emakumeØ etorri da (woman has arrived)

- Predicates in copular sentences require the definite article -a:

  Anne ona da (Anne is good)

  *Anne onØ da (Anne is good)

- A list of indefinite quantifiers (such as zenbait (some); hainbat (many, much); gutxi (few, little); asko (many)) cannot co-occur with any determiner in the same phrase:

  Zenbait gizonØ (some man)

*Zenbait gizona (*some a man)
Hainbat liburuk (many books)
*Hainbat liburuak (*many the books)

These examples show some characteristics of correct and incorrect uses of determiners. Determiner errors are quite common in written Basque, especially in learner corpora, due to the aforementioned morphosyntactic variations and the standardisation process in Basque.

APPENDIX B

XML Schema

```
<?xml version=”1.0” encoding=”UTF−8”?>
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>

<!−− Datatype: string −−>
<xsd:simpleType name=”string.Type”>
  <xsd:restriction base=”xsd:string”/>
</xsd:simpleType>
<!−− Datatype: integer −−>
<xsd:simpleType name=”integer.Type”>
  <xsd:restriction base=”xsd:int”/>
</xsd:simpleType>
<!−− Datatype: float −−>
<xsd:simpleType name=”float.Type”>
  <xsd:restriction base=”xsd:double”/>
</xsd:simpleType>
<!−− Datatype: boolean −−>
<xsd:simpleType name=”boolean.Type”>
  <xsd:restriction base=”xsd:boolean”/>
</xsd:simpleType>
<!−− Datatype: language−−>
<xsd:simpleType name=”language.Type”>
  <xsd:restriction base=”xsd:language”/>
</xsd:simpleType>
<!−− Datatype: uri −−>
<xsd:simpleType name=”uri.Type”>
  <xsd:restriction base=”xsd:anyURI”/>
</xsd:simpleType>

<xsd:simpleType name=”level.Type”>
  <xsd:restriction base=”xsd:string”>
    <xsd:enumeration value=”A1” />
    <xsd:enumeration value=”A2” />
    <xsd:enumeration value=”B1” />
    <xsd:enumeration value=”B2” />
    <xsd:enumeration value=”C1” />
    <xsd:enumeration value=”C2” />
    <xsd:enumeration value=”DBH1” />
    <xsd:enumeration value=”DBH2” />
    <xsd:enumeration value=”DBH3” />
    <xsd:enumeration value=”DBH4” />
    <xsd:enumeration value=”BT1” />
    <xsd:enumeration value=”BT2” />
    <xsd:enumeration value=”UNI” />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name=”subject.Type”>
  <xsd:restriction base=”xsd:string”>
    <xsd:enumeration value=”Basque” />
    <xsd:enumeration value=”Spanish” />
    <xsd:enumeration value=”English” />
    <xsd:enumeration value=”Science” />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name=”type.Type”>
  <xsd:restriction base=”xsd:string”>
    <xsd:enumeration value=”Fill−in−the−blank” />
    <xsd:enumeration value=”Short−answer”/>
    <xsd:enumeration value=”MCQ”/>
    <xsd:enumeration value=”Error−correction”/>
    <xsd:enumeration value=”Word−formation”/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name=”topicGroup.Type”>
  <xsd:restriction base=”xsd:string”>
    <xsd:enumeration value=”A1” />
    <xsd:enumeration value=”A2” />
    <xsd:enumeration value=”B1” />
    <xsd:enumeration value=”B2” />
    <xsd:enumeration value=”C1” />
    <xsd:enumeration value=”C2” />
    <xsd:enumeration value=”DBH1” />
    <xsd:enumeration value=”DBH2” />
    <xsd:enumeration value=”DBH3” />
    <xsd:enumeration value=”DBH4” />
    <xsd:enumeration value=”BT1” />
    <xsd:enumeration value=”BT2” />
    <xsd:enumeration value=”UNI” />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name=”word.Type”>
  <xsd:simpleContent>
    <xsd:extension base=”xsd:string”>
      <xsd:attribute name=”pos” type=”integer.Type” use=”required”/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="analysis.Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="pos" type="integer.Type" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:element name="questions" type="questions.Type"/>

<xsd:complexType name="questions.Type">
  <xsd:sequence>
    <xsd:element name="question" type="question.Type" maxOccurs="unbounded↩
        "/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="question.Type">
  <xsd:sequence>
    <xsd:element name="answer_focus" type="answer_focus.Type" maxOccurs="↩
        unbounded" />
    <xsd:element name="context" type="context.Type"/>
  </xsd:sequence>
  <xsd:attribute name="topic" type="string.Type" use="required"/>
  <xsd:attribute name="level" type="level.Type" use="required"/>
  <xsd:attribute name="source" type="uri.Type" user="required"/>
  <xsd:attribute name="pos" type="integer.Type" use="required"/>
  <xsd:attribute name="type" type="type.Type" use="required"/>
  <xsd:attribute name="language" type="language.Type" use="required"/>
  <xsd:attribute name="subject" type="subject.Type" use="optional"/>
</xsd:complexType>

<xsd:complexType name="answer_focus.Type">
  <xsd:sequence>
    <xsd:element name="head" type="head.Type"/>
    <xsd:element name="notHead" type="notHead.Type"/>
  </xsd:sequence>
  <xsd:attribute name="posQ" type="integer.Type" use="required"/>
  <xsd:attribute name="posS" type="integer.Type" use="required"/>
  <xsd:attribute name="change" type="boolean.Type" use="required"/>
  <xsd:attribute name="blank" type="boolean.Type" use="required"/>
</xsd:complexType>

<xsd:complexType name="head.Type">
  <xsd:sequence>
    <xsd:element name="answer" type="answer.Type"/>
    <xsd:element name="distractor" type="distractor.Type" minOccurs="0" ↩
        maxOccurs="unbounded"/>
    <xsd:element name="headComponent" type="headComponent.Type" minOccurs=↩
        "0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="answer.Type">
```

```
  <xsd:sequence>
    <xsd:element name="word" type="word.Type" maxOccurs="unbounded"/>
    <xsd:element name="topic_info" type="topic_info.Type" />
    <xsd:element name="analysis" type="analysis.Type" maxOccurs="unbounded↩
        "/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="topic_info.Type">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="linguistic_info" type="string.Type"/>
      <xsd:element name="lemma" type="string.Type"/>
    </xsd:choice>
    <xsd:element name="function" type="string.Type" minOccurs="0"/>
    <xsd:attribute name="artificial" type="boolean.Type" use="optional"/>
    <xsd:any minOccurs="0"/>
  </xsd:sequence>
</xsd:element>

<xsd:complexType name="distractor.Type">
  <xsd:sequence>
    <xsd:element name="topicGroup" type="topicGroup.Type" use="required" ↩
        minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="word" type="word.Type" maxOccurs="unbounded"/>
    <xsd:element name="heuristic" type="heuristic.Type"/>
    <xsd:element name="order" type="order.Type" minOccurs="0"/>
    <xsd:element name="analysis" type="analysis.Type" maxOccurs="unbounded↩
        "/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="heuristic.Type">
  <xsd:sequence>
    <xsd:element name="type" type="string.Type"/>
    <xsd:element name="function" type="string.Type"/>
    <xsd:element name="input" type="string.Type"/>
    <xsd:element name="output" type="string.Type"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="order.Type">
  <xsd:sequence>
    <xsd:element name="method" type="string.Type"/>
    <xsd:element name="function" type="integer.Type"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="headComponent.Type">
  <xsd:sequence>
    <xsd:element name="word" type="word.Type" maxOccurs="unbounded"/>
    <xsd:element name="rule" type="string.Type"/>
    <xsd:element name="analysis" type="analysis.Type" maxOccurs="unbounded↩
        "/>
  </xsd:sequence>
  <xsd:attribute name="type" type="string.Type" use="required"/>
</xsd:complexType>
```

```
<xsd:complexType name=”notHead.Type”>
  <xsd:sequence>
    <xsd:element name=”word” type=”word.Type” maxOccurs=”unbounded”/>
    <xsd:element name=”analysis” type=”analysis.Type” maxOccurs=”unbounded↩
        ”/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name=”context.Type”>
  <xsd:sequence>
    <xsd:element name=”chunk” type=”chunk.Type” maxOccurs=”unbounded/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name=”chunk.Type”>
  <xsd:sequence>
    <xsd:element name=”word” type=”word.Type” maxOccurs=”unbounded”/>
    <xsd:element name=”analysis” type=”analysis.Type” maxOccurs=”unbounded↩
        ”/>
    <xsd:element name”function” type=”string.Type” use=”optional”/>
  </xsd:sequence>
  <xsd:attribute name=”posQ” type=”integer.Type” use=”required”/>
  <xsd:attribute name=”posS” type=”integer.Type” use=”required”/>
  <xsd:attribute name=”change” type=”boolean.Type” use=”required”/>
</xsd:complexType>

</xsd:schema>
```

APPENDIX C

Helduen Euskalduntzearen Oinarrizko Kurrikulua

The document "Helduen euskalduntzearen oinarrizko kurrikulua"[1] (HEOK) specifies the Basque language learning process for adults. Within the document, the process of learning the Basque language is divided into four levels. The first level offers a strong basis to the learners, which is then studied in greater depth in the second level. The third level is similar to the level required to obtain the corresponding to level C1 in the Common European Framework. Finally, the last level focuses on professional, specialised and scientific Basque.

Each defined level in HEOK has different objectives and content that are established for different skills: reading, writing, listening and speaking. In this work, we have generated items that take into account the morphosyntactic aspects that learners have to acquire during their learning process in order to deal with the aforementioned skills. Most specifically, we have focused on some declension and verb tenses which appear at each level and must be known by learners. For instance, at the B2 level, learners must know the present indicative and the past conditional tense is not a requirement of this particular level.

Table C.1 presents the declension cases which must be known at each level. Table C.2 specifies the verb tenses.

[1] The basic curriculum for the process of learning the Basque language for adults.

Declension	**SING.**	**PL.**	**IND.**
Absolutive	x	x	x
Ergative	x	x	x
Dative	x	x	x
Inessive	x	x	x
Possessive genitive	x	x	x
Allative	x	x	x
Endpoint allative (-raino)			
Ablative	x	x	x
Comitative	x	x	x
Locative genitive	x	x	x
Benefactive	x	x	x
Motivative	x	x	x
Instrumental			
Partitive			
Prolative			
Two-case complex postpositions: -rako, -ranzko, -rainoko, -tiko, -rekiko, -zko, -rentzako, -renganako, -rengandiko			

Table C.1: Declension cases learnt at each level. Grey: first level; Red: second level; and Blue: third level

Tense	**Paradigm**
Present indicative	NOR: Izan, Egon, Joan, Ibili, Etorri, NOR-NORK: Ukan, Eduki, Jakin, Eraman, Ekarri, Esan (diot...), Iruditu NOR-NORI ZER-NORI-NORK
Past indicative	NOR: Izan, Egon, Joan, Ibili, **Etorri** NOR-NORK: Ukan (nuen...), Eduki, Jakin, **Ekarri, Eraman, Esan** NOR-NORI ZER-NORI-NORK
Present conditional	NOR: Izan, Egon (balego, legoke), **Ibili (balebil)** NOR-NORK: Ukan, Jakin (baneki) **NOR-NORI** **ZER-NORI-NORK**
Past conditional	**NOR** **NOR-NORK** **NOR-NORI** **ZER-NORI-NORK**
Verb idios	NOR: Bizi izan, Ari izan NOR-NORK: Nahi izan, Balio izan, Behar izan, Ahal/Ezin izan
Aspect (present, past)	NOR/NOR-NORK: ez-burutu puntukaria, burutua, ez-burutua, gertakizuna
Aspect of verb expressions	
Imperative mood	NOR: Izan (zaitez, zaitezte), Joan (zoaz), Etorri (zatoz) NOR-NORK: Ukan (ezazu, itzazu, ezazue, itzazue) ZER-NORI-NORK [iezadazu(e), iezaiozu(e)] **ZER-NORI-NORK** **NOR-NORK: [nazazu(e), gaitzazu(e)]** **dadila/dezala** **daitezela/dezatela** **gaitezen/dezagun**
Present subjunctive	**NOR: Izan (nadin...)** **NOR-NORK: Ukan (dezadan...)**
Potential (present)	NOR: Izan NOR-NORK: Ukan **ZER-NORI-NORK**
Potential (past)	**NOR: Izan (nintekeen...)** **NOR-NORK: Ukan (nezakeen...)**
Potential (hypothetic)	**NOR** **NOR-NORK (nezake...)**
Impersonal forms	
Causative verbs: erazi, eragin	
Synthetic verbs: Eritzi, Egokitu	
Dihardut...	
Datza...	
Dario...	
Dirau...	
Darabilt...	
Passive: Partizipioa + A	
Familiar "hi" form	

Table C.2: Learnt verb tenses at each level. Grey: first level; Red: second level; and Blue: third level

APPENDIX D

Determiner Test XML

Heuristics based on determiner errors

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE erroa SYSTEM "/media/DATUAK/Documents/IXA/dete.dtd">
<erroa>
  <etiketa DETE="IZE+DET+ADJ+DET">
    <errore id="OKER_DETE1A_1"/>
    <errore id="OKER_DETE1B_1"/>
    <errore id="OKER_DETE1C_1"/>
    <errore id="OKER_DETE1D1_1"/>
    <errore id="OKER_DETE1D2_1"/>
    <zuzena>ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">ken_IZEDET eta ken_ADJDET</↩
        distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+DET+ART">
    <errore id="OKER_DETE2A_1"/>
    <errore id="OKER_DETE2B_1"/>
    <errore id="OKER_DETE2C_1"/>
    <errore id="OKER_DETE2D1_1"/>
    <errore id="OKER_DETE2D2_1"/>
    <errore id="OKER_DETE2D3_1"/>
    <zuzena>ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00">ald_DZG</distraigarri>
    <distraigarri id="3" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
```

```
</etiketa>
<etiketa DETE="IZE+DET+ADJ+ART">
  <errore id="OKER_DETE3A1_2"/>
  <errore id="OKER_DETE3A2_2"/>
  <errore id="OKER_DETE3D1_2"/>
  <errore id="OKER_DETE3D2_2"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
  <distraigarri id="2" pisua="1.00">ken_IZEDET eta gehi_ADJDET</↩
      distraigarri>
  <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
  <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
      distraigarri>
  <distraigarri id="6" pisua="1.00">ald_DZG</distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
  <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
      distraigarri>
  <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
      distraigarri>
  <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
      distraigarri>
  <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
      distraigarri>
  <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
      distraigarri>
  <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
      distraigarri>
  <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
      distraigarri>
  <distraigarri id="22" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
      distraigarri>
  <distraigarri id="23" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
      distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="25" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
<etiketa DETE="IZE+ADJ+DET+ART">
  <errore id="OKER_DETE3B1_2"/>
  <errore id="OKER_DETE3B2_2"/>
```

```
    <errore id="OKER_DETE3E1_2"/>
    <errore id="OKER_DETE3E2_2"/>
    <zuzena>ken_ADJDET</zuzena>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET eta gehi_IZEDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="1.00">ald_DZG</distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="23" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="25" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+DET+ADJ+DET+ART">
    <errore id="OKER_DETE3C1_2"/>
    <errore id="OKER_DETE3C2_2"/>
    <errore id="OKER_DETE3C3_2"/>
    <errore id="OKER_DETE3F1_2"/>
    <errore id="OKER_DETE3F2_2"/>
    <zuzena>ken_ADJDET eta ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">ken_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="1.00">ald_DZG</distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="23" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="25" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+DET+ERAK">
    <errore id="OKER_DETE4A_1"/>
    <errore id="OKER_DETE4B_1"/>
    <zuzena>ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="IZE+DET+ADJ+ERAK">
    <errore id="OKER_DETE4C1_2"/>
    <errore id="OKER_DETE4C2_2"/>
    <errore id="OKER_DETE4C3_2"/>
    <errore id="OKER_DETE4C3_2"/>
    <errore id="OKER_DETE4F_2"/>
    <zuzena>ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_IZEDET eta gehi_ADJDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+ADJ+DET+ERAK">
    <errore id="OKER_DETE4D1_2"/>
    <errore id="OKER_DETE4D2_2"/>
    <errore id="OKER_DETE4D3_2"/>
    <errore id="OKER_DETE4D4_2"/>
    <errore id="OKER_DETE4G_2"/>
    <zuzena>ken_ADJDET</zuzena>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET eta gehi_IZEDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
  <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
<etiketa DETE="IZE+DET+ADJ+DET+ERAK">
  <errore id="OKER_DETE4E1_2"/>
  <errore id="OKER_DETE4E2_2"/>
  <errore id="OKER_DETE4E3_2"/>
  <errore id="OKER_DETE4E4_2"/>
  <errore id="OKER_DETE4H_2"/>
  <zuzena>ken_ADJDET eta ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ken_IZEDET</distraigarri>
  <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
  <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
  <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
  <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
      distraigarri>
  <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
      distraigarri>
  <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
      distraigarri>
  <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
      distraigarri>
  <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
      distraigarri>
  <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
      distraigarri>
  <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
      distraigarri>
  <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
      distraigarri>
  <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
      distraigarri>
  <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
```

```
<etiketa DETE="IZE+DET+ORO">
  <errore id="OKER_DETE6A1_1"/>
  <errore id="OKER_DETE6A2_1"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="2" pisua="1.00">ald_DZG</distraigarri>
  <distraigarri id="3" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
  <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
</etiketa>
<etiketa DETE="IZE+DET+DZG">
  <errore id="OKER_DETE7A_1"/>
  <errore id="OKER_DETE7B_1"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
  <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
</etiketa>
<etiketa DETE="DZG+IZE+DET">
  <errore id="OKER_DETE9A_1"/>
  <errore id="OKER_DETE9B_1"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
```

```
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="DZG+IZE+ADJ+DET">
    <errore id="OKER_DETE9C_2"/>
    <zuzena>ken_ADJDET</zuzena>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET eta gehi_IZEDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
  <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
      distraigarri>
  <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
      distraigarri>
  <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
<etiketa DETE="DZG+IZE+DET+ADJ+DET">
  <errore id="OKER_DETE9D_2"/>
  <zuzena>ken_ADJDET eta ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ken_IZEDET</distraigarri>
  <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
  <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
  <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
  <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
      distraigarri>
  <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
      distraigarri>
  <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
      distraigarri>
  <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
      distraigarri>
  <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
      distraigarri>
  <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
      distraigarri>
  <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
      distraigarri>
  <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
      distraigarri>
  <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
      distraigarri>
  <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
```

```
<etiketa DETE="DZG|NOLGAL|NOLARR+IZE+DET">
  <errore id="OKER_DETE10A_1"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
  <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
</etiketa>
<etiketa DETE="DZG|NOLGAL|NOLARR+IZE+DET+ADJ+DET">
  <errore id="OKER_DETE10B_2"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ken_IZEDET eta ken_ADJDET</↩
      distraigarri>
  <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
  <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
  <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
  <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
      distraigarri>
  <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
      distraigarri>
  <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
      distraigarri>
  <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
      distraigarri>
  <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
      distraigarri>
  <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
      distraigarri>
  <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
      distraigarri>
  <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
      distraigarri>
  <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
      distraigarri>
  <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
  <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
</etiketa>
<etiketa DETE="ZBKI+IZE+DET">
  <errore id="OKER_DETE11_1"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
      gehi_IZEKAS_berdina</distraigarri>
  <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
      distraigarri>
  <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
      distraigarri>
</etiketa>
<etiketa DETE="ZBKI+IZE+DET+ADJ">
  <errore id="OKER_DETE12A_2"/>
  <zuzena>ken_IZEDET</zuzena>
  <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
  <distraigarri id="2" pisua="1.00">ken_IZEDET eta gehi_ADJDET</↩
      distraigarri>
  <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
  <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
  <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
      distraigarri>
  <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
      distraigarri>
  <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
      distraigarri>
  <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
      distraigarri>
  <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
      distraigarri>
```

```
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="ZBKI+IZE+ADJ+DET">
    <errore id="OKER_DETE12B_2"/>
    <zuzena>ken_ADJDET</zuzena>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET eta gehi_IZEDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="ZBKI+IZE+DET+ADJ+DET">
    <errore id="OKER_DETE12C_2"/>
    <errore id="OKER_DETE12D_2"/>
    <errore id="OKER_DETE12E_2"/>
    <zuzena>ken_ADJDET eta ken_IZEDET</zuzena>
    <distraigarri id="1" pisua="1.00">ken_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
</erroa>
```

Heuristics based on correct answers

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE erroa SYSTEM "/home/olatz/programak/zuzdistr.dtd">
<erroa>
  <etiketa DETE="IZE+ADJ+DET">
    <zuzena id="ZUZEN1_1"></zuzena>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET eta ken_ADJDET</↩
        distraigarri>
    <distraigarri id="2" pisua="1.00">ken_ADJDET</distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+ART">
    <zuzena id="ZUZEN2_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00">ald_DZG</distraigarri>
    <distraigarri id="3" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="IZE+ADJ+ART">
    <zuzena id="ZUZEN3_2"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="2" pisua="1.00">gehi_IZEDET eta gehi_ADJDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id=”6” pisua=”1.00”>ald_DZG</distraigarri>
    <distraigarri id=”7” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id=”8” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id=”9” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id=”10” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id=”11” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id=”12” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id=”13” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id=”14” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id=”15” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id=”16” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id=”17” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id=”18” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id=”19” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id=”20” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id=”21” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id=”22” pisua=”0” baldintza=”abs_bai”>gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id=”23” pisua=”0” baldintza=”abs_ez”>gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id=”24” pisua=”1.00” baldintza=”abs_ez”>↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id=”25” pisua=”1.00” baldintza=”abs_ez”>↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE=”IZE+ERAK”>
    <zuzena id=”ZUZEN4_1”/>
    <distraigarri id=”0” pisua=”1.00”>gehi_IZEDET</distraigarri>
    <distraigarri id=”1” pisua=”1.00”>ald_IZENUM</distraigarri>
    <distraigarri id=”2” pisua=”1.00” baldintza=”abs_ez”>↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id=”3” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id=”4” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id=”5” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id=”6” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id=”7” pisua=”0” baldintza=”abs_bai”>gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="IZE+ADJ+ERAK">
    <zuzena id="ZUZEN5_2"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="2" pisua="1.00">gehi_IZEDET eta gehi_ADJDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="IZE+ORO">
```

```
    <zuzena id="ZUZEN6_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00">ald_DZG</distraigarri>
    <distraigarri id="3" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="IZE+DZG">
    <zuzena id="ZUZEN7_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="DZG+IZE">
    <zuzena id="ZUZEN8_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="DZG+IZE+ADJ">
    <zuzena id="ZUZEN9_2"/>
    <distraigarri id="0" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="2" pisua="1.00">gehi_ADJDET eta gehi_IZEDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="DZG|NOLGAL|NOLARR+IZE">
    <zuzena id="ZUZEN10_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="DZG|NOLGAL|NOLARR+IZE+ADJ">
    <zuzena id="ZUZEN11_2"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_IZEDET eta gehi_ADJDET</↩
        distraigarri>
    <distraigarri id="2" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
  <etiketa DETE="ZBKI+IZE">
    <zuzena id="ZUZEN12_1"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="2" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
    <distraigarri id="3" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="4" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="5" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
  </etiketa>
  <etiketa DETE="ZBKI+IZE+ADJ">
    <zuzena id="ZUZEN13_2"/>
    <distraigarri id="0" pisua="1.00">gehi_IZEDET</distraigarri>
    <distraigarri id="1" pisua="1.00">gehi_ADJDET</distraigarri>
    <distraigarri id="2" pisua="1.00">gehi_IZEDET eta gehi_ADJDET</↩
        distraigarri>
    <distraigarri id="3" pisua="1.00">ald_ADJNUM</distraigarri>
    <distraigarri id="4" pisua="1.00">ald_IZENUM</distraigarri>
    <distraigarri id="5" pisua="1.00">ald_ADJNUM eta ald_IZENUM</↩
        distraigarri>
    <distraigarri id="6" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ALA</↩
        distraigarri>
    <distraigarri id="7" pisua="0" baldintza="abs_bai">gehi_IZEKAS_SOZ</↩
        distraigarri>
    <distraigarri id="8" pisua="0" baldintza="abs_bai">gehi_IZEKAS_DAT</↩
        distraigarri>
    <distraigarri id="9" pisua="0" baldintza="abs_bai">gehi_IZEKAS_ERG</↩
        distraigarri>
    <distraigarri id="10" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEL</↩
        distraigarri>
    <distraigarri id="11" pisua="0" baldintza="abs_bai">gehi_IZEKAS_GEN</↩
        distraigarri>
    <distraigarri id="12" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INE</↩
        distraigarri>
    <distraigarri id="13" pisua="0" baldintza="abs_bai">gehi_IZEKAS_INS</↩
        distraigarri>
    <distraigarri id="14" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ALA</↩
        distraigarri>
    <distraigarri id="15" pisua="0" baldintza="abs_bai">gehi_ADJKAS_SOZ</↩
        distraigarri>
    <distraigarri id="16" pisua="0" baldintza="abs_bai">gehi_ADJKAS_DAT</↩
        distraigarri>
    <distraigarri id="17" pisua="0" baldintza="abs_bai">gehi_ADJKAS_ERG</↩
        distraigarri>
    <distraigarri id="18" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEL</↩
        distraigarri>
    <distraigarri id="19" pisua="0" baldintza="abs_bai">gehi_ADJKAS_GEN</↩
        distraigarri>
    <distraigarri id="20" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INE</↩
        distraigarri>
    <distraigarri id="21" pisua="0" baldintza="abs_bai">gehi_ADJKAS_INS</↩
        distraigarri>
    <distraigarri id="22" pisua="0" baldintza="abs_ez">gehi_ADJKAS_ABS</↩
        distraigarri>
    <distraigarri id="23" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina eta gehi_ADJKAS_ABS</distraigarri>
    <distraigarri id="24" pisua="1.00" baldintza="abs_ez">↩
        gehi_IZEKAS_berdina</distraigarri>
  </etiketa>
</erroa>
```