

Question Generation Based on Numerical Entities in Basque

Itziar Aldabe, Montse Maritxalar and Ander Soraluze

IXA NLP Research Group
University of the Basque Country
Tolosa Etorbidea 72, Donostia-San Sebastian
itziar.aldabe@ehu.es

Abstract

This article presents a question generation (QG) system which is integrated within an automatic exercise generation system. The QG system deals with Basque language and the target selection is restricted to numerical entities. In this article we present an experiment which was conducted on a specialised corpus on science and technology and the system was evaluated manually and automatically.

Motivation

ArikIturri (Aldabe et al. 2006) is a system developed for the automatic generation of different types of exercise. 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. Thus, the system is able to automatically generate tests from texts, to be included in testing tasks. The system is able to produce fill-in-the-blank (FBQ), word formation, multiple-choice question (MCQ), error correction and short answer questions. In previous studies (Aldabe et al. 2006; Aldabe, Maritxalar, and Mitkov 2009), we demonstrate the viability of the system to generate items in two languages: Basque and English. Figure 1 shows the main modules of its architecture.

In our approach, the stem of an item is selected from the input corpus. For us, 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. The source sentence has to include the topic 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. For this purpose, we have implemented and integrated a question generation (QG) system within ArikIturri.

In fact, this is a challenging task that has attracted a number of researchers over the last years. In 2008, the first Workshop on Question Generation (Rus and Graesser 2009) was

held and this was the starting point for this ever-increasing community. QG is defined (Rus and Graesser 2009) as the task of automatically generating questions from some form of input.

Among the various tasks which have been proposed (Rus and Graesser 2009), 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. This task contains three steps: *Target Selection*; *Question Type Selection*; and *Question Construction*. In brief, first, in the *Target Selection* step, the topic is identified.¹ 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 our QG system, the sentence retriever module is responsible for the *Target Selection* task and the item generator module performs the *Question Type Selection* and *Question Construction* processes.

As regards the evaluation measures, (Rus and Graesser 2009) mention that QG systems can be evaluated either manually or automatically. In this article, we explore both options.

This article presents a QG system and its evaluation results. For this purpose, we focused on a specialised corpus, the ZT corpus, which is a collection of Basque texts relating to science and technology. The experiment created questions (interrogative stems) regarding numerical entities. The evaluation focused on how well the system creates the corresponding wh-word² based on the linguistic 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. Finally, some conclusions and future work are outlined.

Question generation

Our QG system 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 is conceived as a shallow question generator which deals with the Basque language. Although it aims

¹This is also referred to as the Key Concept Identification task.

²In this article, we refer to interrogative words as wh-words.

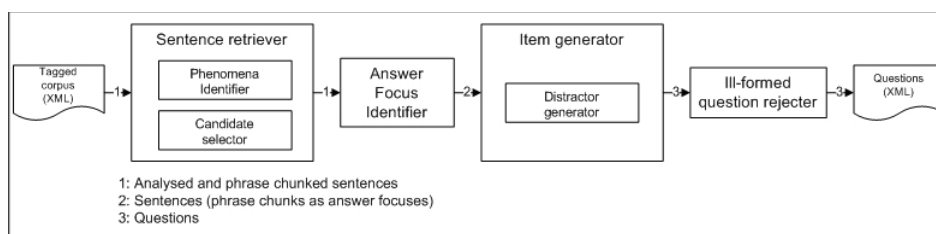


Figure 1: The architecture of ArikIturri

to be a complete system in which the three different steps are fulfilled 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, a specialised corpus on science and technology.

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

In the *Target Selection* task, 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, through which WHICH, HOW MANY and WHEN wh-words were identified. Finally, in the *Question Generation* task, some transformation rules are proposed in order to modify the source information to obtain the corresponding question.

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 during the workshops on QG 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).

In contrast, 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. In our approach, once the clauses have been identified, the identification of numerical entities is conducted.

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 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, Màrquez, and Castro 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.

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%.³ 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.⁴ 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 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, the QG system detected two coordinated clauses:

Joan den abenduan argitaratu zuen txostena (The report was published last December)

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

³The same baseline system for English achieves a score of 47.71%.

⁴Rule-based grammar was originally used to tag noun and verb chains as well as sentences and clauses.

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.

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* (Aduriz et al. 2004), a robust cascaded syntactic analyser for linguistic analysis in Basque. The analyser identifies and tags numbers according to six predefined types:

ZEN: used to mark non-declined numbers written with digits; cardinals 22; percentages % 4,5; times 23:30; etc.

ZEN_DEK: used for declined numbers; Declined numbers; cardinals 22k (22), 45i (to 45), 5ek (the 5); percentages % 45etan (in 45%); times 23:30etan (at 23:30), etc.

HAUL_ZNB: used for multiword numbers; 98 milioi (98 million);

HAUL_DATA: used when a multiword date structure is detected; martxoaren 19an (on March 19th);

ERROM: used for Roman numerals; VI;

DET_DZH: used for numbers written in characters; hamaika (eleven).

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, authors distinguish between percentage expressions like *hazkundera % 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 **hazkundera** (**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). 34 FSTs to classify closed categories and two more for open categories that refer to common nouns have been defined. The rules were defined using *foma*⁵ (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.⁶

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

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 (see Figure 2).

(Euria ari zuen arren,) oinez joan ginen.)
 ((Although it was raining,) we went on foot.)

Figure 2: Recursive representation of a sentence

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

⁵*foma* is an opensource toolkit.

⁶Those are two well-known evaluation methods.

	Pattern	wh-word
1	((IZE ZEN or DET DZH))	ZENBAT (HOW MANY)
2	((IZE ZEN or DET DZH) and ERG)	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)	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
34	DET ORD and INE	ZENBAGARRENEAN (IN WHICH)
35

Table 1: Patterns for recognising numerical entities⁷

((Joan den) abenduan argitaratu zuen txostena) eta (ot-sailaren 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 the previous source sentence, the system proposes as candidate clauses, among others, *otsailaren 25a arte nahi due-*

⁷IZE: noun; ZEN: number; DET DZH: numbers written in characters. ERG: ergative; DAT: dative; GEN: genitive; SOZ: sociative; DES: destinative; INS: instrumental; INE: inessive; ALA: allative; ABU: abulative; ABZ: and GEL: locative genitive. BIZ: animate. ZENBATERAKO, ZENBATENGAN, ZENBATENGANDIK, ZENBATENGANA, ZENBATENGANANTZ, ZENBATENGANAINO and ZENBATENGATIK wh-words asked about animates.

nak,⁸ *otsailaren 25a arte nahi duenak iritzia emateko*⁹ and *otsailaren 25a arte nahi duenak iritzia emateko aukera du*.¹⁰

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 produce any candidate sentence. In contrast, in this order, the system chooses as a candidate *otsailaren 25a arte nahi duenak iritzia emateko aukera du*.

Question type identification

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 46 patterns to recognise first

⁸Until February 25th those who want to do so

⁹Until February 25th, those who want to do so to express their views

¹⁰Until February 25th, those who want to do so have the opportunity to express their views

the numerical entities and then the morphosyntactic information in order to establish the corresponding wh-words. Table 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 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 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 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 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.lean (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 25th)*, the chunk that corresponds to it is *Otsailaren 25a arte (until February 25th)*. It is a date magnitude, and the last compo-

nent of the date expression contains the allative case. Therefore, the corresponding wh-word is **NOIZ ARTE (UNTIL WHEN)**.

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 the source sentence, the system generates the question following displayed:

NOIZ ARTE du nahi duenak iritzia emateko aukera?

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

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.¹¹ 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 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.

This evaluation was carried out by one human rater and Table 3 summarises the results obtained. The results

¹¹In the given data, for each sentence, organisers provide the question types that can be generated.

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 changes
3	There are some grammatical errors in the question	Major changes
4	The question is grammatically unacceptable	Discard

Table 2: Scoring for syntactic correctness and fluency

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.

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

Table 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. 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 can-

didate 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.

Conclusions and future work

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.

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. 2003). 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. The research line started by (Chen, Aist, and Mostow 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.

Acknowledgments.

We would like to thank Ainara Estarrona for the manual evaluation. This work has been supported by Ander Soraluze's PhD grant from the University of the Basque Country (UPV/EHU) and the BERBATEK project (IE09-262).

References

- Aduriz, I.; Aranzabe, M.; Arriola, J.; Díaz de Ilarraza, A.; Gojenola, K.; Oronoz, M.; and Uria, L. 2004. A cascaded syntactic analyser for Basque. *Computational Linguistics and Intelligent Text Processing* 124–135.
- Aldabe, I.; de Lacalle, M. L.; Maritxalar, M.; Martinez, E.; and Uria, L. 2006. Arikiturri: An automatic question generator based on corpora and NLP techniques. *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems (ITS'06)* 584–594.
- Aldabe, I.; Maritxalar, M.; and Mitkov, R. 2009. 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)*, 656–658.
- Aldezabal, I.; Aranzabe, M.; Díaz de Ilarraza, A.; Estarrona, A.; and Uria, L. 2010. EusPropBank: Integrating semantic information in the Basque dependency treebank. In Gelbukh, A., ed., *Computational Linguistics and Intelligent Text Processing*, volume 6008/2010 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 60–73.
- Alegria, I.; Ezeiza, N.; Fernandez, I.; and Urizar, R. 2003. 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*.
- Alegria, I.; Arrieta, B.; Carreras, X.; Díaz de Ilarraza, A.; and Uria, L. 2008. Chunk and clause identification for Basque by filtering and ranking with perceptrons. In *Proceedings of SEPLN*.
- Becker, L.; Nielsen, R.; Okoye, I.; Sumner, T.; and Ward, W. 2010. What's next? Target concept identification and sequencing. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 35–44.
- Boyer, K., and Piwek, P., eds. 2010. *Proceedings of QG2010: The Third Workshop on Question Generation*. Pittsburgh: questiongeneration.org.
- Carreras, X.; Márquez, L.; and Castro, J. 2005. Filtering–ranking perceptron learning for partial parsing. *Machine Learning* 60:41–71.
- Chen, W.; Aist, G.; and Mostow, J. 2009. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation*, 17–24.
- Heilman, M., and Smith, N. 2010. Extracting simplified statements for factual question generation. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 11–20.
- Hulden, M. 2009. Foma: a finite-state compiler and library. In *Proceedings of EACL 2009*, 29–32.
- Rus, V., and Graesser, A., eds. 2009. *The Question Generation Shared Task and Evaluation Challenge*. Sponsored by the National Science Foundation. ISBN 978-0-615-27428-7.
- Soraluze, A.; Alegria, I.; Ansa, O.; Arregi, O.; and Arregi, X. 2011. Recognition and classification of numerical entities in Basque. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*.