Making Numerical Information more Accessible: Implementation of a Numerical Expression Simplification System for Spanish

Susana Bautista
NIL: Natural Interaction based on Language
Universidad Complutense de Madrid
C/ Prof. José García Santesmases, s/n.
28040 Madrid, Spain

Horacio Saggion
TALN: Natural Language Processing Group
Universitat Pompeu Fabra
C/Tanger 122
08018 Barcelona, Spain

Abstract

Are rounded numbers easier to understand than exact numbers? Information in newspapers often takes the form of numerical expressions which pose comprehension problems for many people, including people with disabilities, illiteracy or lack of access to advanced technology. The purpose of this paper is to motivate and describe a rule-based lexical component that simplifies numerical expressions in Spanish texts. We propose a simplification approach that makes news articles more accessible to readers with specials needs by rewriting difficult numerical expressions in a simpler way. We carry out a study that identifies simplification strategies used to simplify numerical information in the text by analyzing a parallel corpus of original texts and their manual simplifications. The study is complemented with an analysis of simplifications obtained in response to a questionnaire where subjects were asked to produce simplifications of numerical expressions in context. We have implemented and evaluated a simplification system that mimics the simplification strategies discovered.

Key words: Simplification Corpus, Numerical Expressions, Simplification Strategies, Text Accessibility

1. Introduction

Access to information is a fundamental human right which was asserted in the United Nations' Universal Declaration of Human Rights. However, making information universally available through democratic channels such as the Internet does not guarantee universal accessibility to information because the way in which information is written or presented has a great impact on text readability and comprehension. Thus texts which are understandable for the general public could be difficult to understand for specific groups of people. According to Snow (2002), reading comprehension entails three elements: the reader who is meant to comprehend, the text that is to be comprehended, and the activity in which comprehension takes part. In addition to the content presented in the text, its vocabulary load, its linguistic structure, discourse style, the quantity of numerical information and genre, they all interact with the reader's knowledge and have an influence on reading comprehension too. When these factors do not match the reader's knowledge and experience, the text becomes too complex for appropriate comprehension.

In order to make a text more understandable for a target population the factors that make the text complex should be assessed and the text simplified accordingly. A number of guidelines have been proposed to make a text "easy to read and understand"— see for example Plain Language (2005), the European Guidelines for the Production of Easy-to-Read Information (Freyhoff, G., Hess, G., Kerr, L, Menzel, E., Tronbacke, B. and Veken, K.V.D., 1998) or the Web Content Accessibility Guidelines (W3C, 2008). However, general guidelines would probably not be enough to address specific user groups or text difficulty phenomena. Adapting texts to make them easy to understand for specific target user groups is generally done manually, making massive production of easy-to-read texts practically impossible. Automatic text simplification is

a technology which has the potential to speed up the process of transforming a text into a quasi equivalent which could be more understandable. In addition to the topic of the text, which in itself can be a burden to the reader, stylistic and grammatical features of the text can impact comprehension too. Long sentences, coordination, passive constructions, embedded clauses, non-canonical word order, use of low frequency words, abundance of numerical information, among other things, have been recognised as factors which may make a text difficult to understand (Siddharthan, 2002) (Klevanov, B.B., Knight, K. and Marcu, D., 2004) (Devlin, S. and Unthank, G., 2006) (Bautista, S., Gervás, P. and Madrid, R.I., 2009) (Caseli, H.M., Pereira, T.F., Specia, L., Pardo, T.A.S, Gasperin, C. and Aluisio, S.M., 2009).

One important type of information which occurs independently of the text genre and which is particularly abundant in newspaper articles is numerical information, be it dates, measurements, quantities, percentages, or ratios. Numerical information poses comprehension problems for different populations and in particular for people with limited education. Indeed, recent surveys of literacy in the UK (Deeqa Jama, 2010) (Department of Education, 2003) (Clark, 2010) revealed that 7 million adults in England cannot locate the reference page for plumbers if given the Yellow Pages alphabetical index, meaning that one in five adults in England has a lower literacy level than what is expected of a 11-year-old child.

In this paper we address a very specific problem in text simplification: the transformation of numerical expressions in text into quasi equivalents which are easy to read and understand. We define a “numerical expression” as a phrase that consists of a quantity, sometimes modified by a numerical hedge (modifier) and some units. The

logical order is the optional hedge, the quantity and the optional units, as in these examples: “around ½ kg”, “less than a quarter”, “about 97.2%”, “almost 100 kms”, “just over 28.5%”. The work is carried out in the context of a project on text simplification for the Spanish language (Saggion, H., Gómez-Martínez, E., Esteban Etayo, E., Anula, A. and Bourg, L., 2011), which aims at making news texts more readable and comprehensible for people with cognitive disabilities. We have proposed a numerical simplification component that transforms a text into a simpler version by reducing the complexity of the numerical information expressed therein. To the best of our knowledge this is the first numerical simplification system for the Spanish language. The simplification system also targets and reduces the lexical and syntactic complexity of Spanish texts; however in this paper we concentrate on this innovative aspect of making numerical information more accessible.

In order to address the problem of numerical expression simplification, we require a set of rewriting strategies to transform a difficult numerical expression into a simpler “equivalent” which is linguistically correct. For example, the expression ‘25.9%’ could be rewritten as ‘just over a quarter’, preserving the intended meaning and losing only a bit of precision. Loss of precision is not necessarily a bad thing, for several reasons. Loss of precision can be signalled linguistically by numerical hedges such as ‘around’, ‘more than’ and ‘a little under’, so it need not be misleading. It is worth noting that numerical expression simplification is a normal practice in newspaper article editing and an important summarization operation. It is indeed not uncommon to see headlines with vague numerical expressions (“Calcutta Fire Kills Dozens”) corresponding to more precise information (“At least 89 people have been killed in a fire in the Indian city of Calcutta...”).

As Krifka (2002) has argued, competent writers and speakers frequently approximate numerical information, and readers and hearers can readily recognise this, even when no hedge is present, especially when numbers are round. For instance, in 'the distance from Oxford to Cambridge is 100 miles' it is clear that 100 miles is an approximation. According to Krifka, an inappropriately high level of precision would violate Grice's Maxim of Quantity (Grice, 1975) by giving too much information. Williams and Power (2009) showed that writers tend to approximate numerical quantities early in a document, and then later in the document give more precise versions of these.

In this paper we present a rule-based numerical rewriting component that transforms numerical expression in Spanish into their simpler equivalents. Our work is grounded on the literature on complexity of numerical expressions, a corpus study that shows how sentences containing numerical expressions are transformed in order to make them easier to understand, and on the analysis of answers collected from a survey with subjects who were asked to simplify numerical expressions in context. The remainder of the article is organized as follows: in Section 2 we present an overview of the most relevant work in the field of automatic text simplification. In Section 3 we outline our approach to the task at hand, describe in some detail the methodology followed, and present our rule-based lexical component. Section 4 discusses the approach and we conclude the article with some conclusions and plans for future work in Section 5.

2. Related work

Text simplification, a relatively new task in Natural Language Processing (Chandrasekar, R., Doran, C. and Srinivas, B., 1996) (Devlin, S and Tait, J., 1998) has

been directed mainly at reducing lexical and syntactic complexity of texts. Working systems and prototypes can be studied along three main lines: the method used by the system (either rule-based or trainable), the type of lexical and syntactic knowledge used to simplify the text, and finally the system's goal.

A number of rule-based systems have been developed over the years (Chandrasekar, R., Doran, C. and Srinivas, B., 1996) (Siddharthan, A., 2003), (Bautista, S., Gervás, P. and Madrid, R.I., 2009) (Aluísio, S. M., Specia, L., Pardo, T.A., Maziero, E. and Fortes, R., 2008) focusing on different kinds of readers (e.g., subjects with low literacy levels or aphasic people). These systems contain a set of manually created simplification rules that are applied to each sentence where each rule matches a particular target syntactic construction (e.g. coordination, relative) and fires appropriate transformation procedures. Siddharthan (2003) proposes a syntactic simplification architecture that relies on shallow text analysis and favours time performance. The general goal of the architecture is to make texts more accessible to a broader audience without selecting a specific target group.

Max (2006) applies text simplification directly during the writing process by embedding an interactive text simplification system into a word processor. At the user's request, an automatic parser analyzes an individual sentence and the system applies handcrafted rewriting rules. This system requires human intervention at every step.

The transformation of texts into easy-to-read versions can be seen as a translation problem between two different subsets of language: the original complex or normal language and an easy-to-read kind of language. Corpus-based systems can learn from

corpora the relevant simplification operations and also the required degree of simplification for a given task (Zhu, Z., Bernhard, D. and Gurevych, I., 2010) (Specia, 2010). Petersen and Ostendorf (2007) address the task of text simplification in the context of second language learning. A data-driven approach to simplification is proposed based on the study of a corpus of paired articles in which not every original sentence necessarily has a corresponding simplified sentence, making it possible to learn where writers have dropped or simplified sentences. A classifier is used to select the sentences to simplify, and Siddharthan's syntactic simplification system is used to split the selected sentences.

There is a variety of problems that researchers on text simplification have addressed, including substituting difficult words for their simpler equivalents (Devlin, S and Tait, J., 1998) (Specia, L, Jahuar, S.K. and Milhacea, R., 2012) (Bott, S., Rello, L., Drndarevic, B. and Saggion, H., 2012), transforming passive into active sentences and resolving coreferences (Canning, 2000), reducing multiple-clause sentences to single-clause sentences (Chandrasekar, R. and Srinivas, B., 1997) (Canning, 2000) (Siddharthan, 2002), and making appropriate choices at the discourse level (Williams, S., Reiter, E. and Osman, L.M., 2003).

Previous works on numerical expressions have studied the treatment of numerical information for experts in different areas, such as health care (Peters, E., Hibbard, J., Slovic, P. and Dieckmann, N., 2007), forecast (Dieckmann, N., Slovic, P. and Peters, E., 2009), representation of probabilistic information (Bisantz, A.M., Schinzing, S. and Munch, J., 2005), or vague information (Mishra, H., Mirshra, A. and Shiy B., 2011).

The issue of simplifying numerical information has been only scarcely studied. Bautista et al. (2011) and Power and Williams (2012) are among the first to focus on the possibility of simplifying this kind of expressions, concentrating mainly on the use of modifiers. Power and Williams (2012) carried out a corpus study of English news, analyzing how the mathematical forms were used by the authors and how precision changed. In a document, the same quantity was described in different ways, using different mathematical forms (fraction, percentage), applying modifiers and rounding operations to account for loss of precision. The task of numerical simplification was addressed as a constraint satisfaction problem, with solutions subsequently ranked by preferences. In Bautista et al. (2011) an analysis was carried out to discover preferences for simplified numerical values. Results showed marked preferences for common numerical values when numbers have to be rounded (e.g. to simplify 50.8% they prefer 50% instead of 51%). These findings were implemented in a customizable numerical expression simplification system for English (Bautista, S., Hervás, R., Gervás, P., Power, R. and Williams, S., 2013). The system for Spanish presented in this paper greatly extends the functionalities of the English system by incorporating different types of numerical expressions.

Text simplification in Spanish has only recently been addressed and a prototype of a system based on a rule-based lexical transformation component and a syntactic simplification module has been developed and evaluated for simplicity and meaning preservation (Drndarevic B, Stajner S, Bott S, Bautista S and Saggion H., 2013). Where simplification of numerical expressions in Spanish is concerned, a parallel corpus of original and manually simplified texts was compiled, with the aim of developing a rule-based system (Bautista, S., Drndarevic, B., Hervás, R., Saggion, H. and Gervás, P.,

2012). However, to the best of our knowledge, the component to be described in this paper is the first developed and evaluated for the Spanish language.

3. Research Methodology

Our methodology consists in: (1) developing and evaluating a component for the identification of numerical expressions in Spanish texts; (2) analysing a parallel corpus of original and manually simplified news articles, aimed at extracting types of simplification operations to be automated; (3) analysing the answers collected from a survey, which asked subjects to simplify numerical expressions from the corpus and upon which we identified simplification strategies used; (4) building a rule-based lexical component for automatic simplification of numerical expressions; and (5) evaluating the automatically simplified output.

In order to ground our approach we relied on a parallel corpus of original and simplified documents. This corpus consists of international and cultural news articles in Spanish, provided by the Spanish news agency Servimedia[1]. Table 1 shows the number of sentences, number of tokens, average sentences per document and average tokens per sentence in the corpus. From the Project Simplext (2011), we selected a set of 40 original and manually simplified news articles in Spanish. Simplifications had been produced by trained human editors, aware of the needs of a person with cognitive disabilities and following a series of easy-to-read guidelines suggested by Anula (2007). It is important to note that these guidelines do not concern the treatment of numerical expressions specifically. This corpus has been used in all the steps described hereafter.

[Here Table 1]

[1] http://www.servimedia.es/

3.1. Identifying Numerical Expressions in Spanish

With the aim of developing a rule-based lexical component to simplify numerical expressions, we create the first specialized component to identify numerical expressions in Spanish texts. We base its development on two widely used tools for Natural Language Processing research: FreeLing (Padró, Ll., Collado, M., Reese, S., loberes, M. and Castelln, I., 2010), the best known system for linguistic treatment of Spanish, and the General Architecture for Text Engineering (GATE) (Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K. and Wilks, Y., 2002), which provides support for corpus analysis and development of simplification rules through its JAPE engine. We use the two components separately: FreeLing is used to analyse a document in order to produce tokens, sentences, and parts-of-speech tags from which we create an XML representation. The latter can be used within the GATE system directly. FreeLing uses a Hidden Markov Approach to tagging, producing an EAGLES tag for each word in the document. Since we are mainly interested in numerical information, we concentrate on tags of type Z which are allocated to quantities, ratios, fractions, percentages, etc. Four different kinds of Z tags are identified: (1) partitive numerals which have the tag Zd (for example, a million (un millón), a hundred (una centena), etc.). (2) monetary expressions having the tag Zm, their lemma being the quantity and the monetary unit (for example, 2000 dollars (2000 dólares), where the lemma is $USD: 2000), (3) fractions and percentages, which are allocated the tag Zp, where the lemma is a normalized proportion (for example, 34% its lemma is 34/100), and finally (4) physical measures having the tag Zu. Their lemma is a normalized notation of the unit and the quantity (for example, 30km/h, with the lemma SP km/h:30).

3.3. Automatic Annotation of Numerical Expressions

In order to develop the numerical expression recognizer we rely on the Java Annotation Pattern Engine (JAPE), a regular expression recognizer implemented in GATE. We define a series of JAPE grammars in order to tag the different types of numerical expressions in the original texts, with their possible modifiers. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern and action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consists of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements. The LHS is the part preceding the '-->' and the RHS is the part following it. The LHS specifies a pattern to be matched to the annotated GATE document, whereas the RHS specifies what is to be done to the matched text.

In the next example, we have a rule called “CasiPorcFract”, which identifies a percentage or a fraction preceded by a modifier. The pattern on the LHS of the rule indicates that text annotated with a “word” tag whose “lemma” feature has the value “casi” (i.e. “almost”) and is followed optionally by another word, needs to be matched. In addition, it will match any text annotated with a “word” annotation whose “tag” feature has the value “Zp”. Examples of this kind of numerical expressions are: “casi 40% (almost 40%)”, “casi el 20% (almost 20%)” or “casi un cuarto (almost a quarter)”.

```
Rule: CasiPorcFract
(((word.lemma==”casi") (word)?): modifier
```

(word.tag==”Zp")):annotate

-->

:modifier.MOD EXP={semantics=”casi"},

:annotate.CASIporcFract= {semantics=”porcFract"}

Once this rule has matched a sequence in the text, the entire sequence is allocated a label by the rule. Additionally, different parts which have been matched against the pattern can be associated to labels we need for specific treatment during simplification. The span of the text covered with the label “modifier” on the RHS is referred to using the same label on the LHS. We say that this text is to be given an annotation of type “MOD_EXP” and a “semantics” feature set to “casi” for the “modifier” label. The other label on the RHS, “annotate”, is used to say that the text covered is to be given an annotation of type “CASIporcFract” and a “semantics” feature se to “porcFract”. We are mainly concerned with the local contextual information of the numerical information that FreeLing provides, since based on the context, different simplification strategies would be possible. In Figure 1 we can see an example of original text from the corpus with numerical expressions identified.

[Here Figure 1]

We use these grammars to recognize different kinds of numerical expressions which we need to manually analyse in order to understand how they are simplified. For analysing the corpus we have relied on the ANNIC system (Aswani, N., Tablan,V., Bontcheva, K. and Cunningham, H., 2005), which allows us to see annotations in context. This system lets us make searches in the tagged corpus with labels generated from the defined rules in our grammars and improve rule coverage though an iterative cycle. The final grammars, which contain 45 different rules, cover 13 different cases of numerical

expressions corresponding to the 4 numerical tags identified by FreeLing. Table 2 presents the identified types which cover all numerical information in the analysed corpus.

[Here Table 2]

We have applied the developed rules to a subset of 10 unseen documents from the Simplext corpus comprising 59 sentences. An analysis of the results indicates that some specific numerical expressions were not covered by our rules therefore prompting the addition of 3 new tags defined to identify additional numerical constructs: "DURANTENUM" to tag expressions like "for 2 hours (durante 2 horas)", "MASDEPART" to tag expressions like "more than 3 million (más de 3 millones)" and "CASIporcFract" to tag expressions like "almost a third (casi un tercio)". By adding these new tags and rules, we corrected the automatic annotations produced by the system. We tested the performance of the rules and obtained a precision of 0.94, a recall of 0.93, and an F-measure of 0.93, which we consider quite acceptable. For numerical expressions with low frequency of occurrence, the results are worse but they are better for frequently observed numerical expressions.

3.3. Analysis of Simplifications in the Corpus

The Simplext corpus consisting of pairs of original and simplified documents was aligned at the sentence level, first using an unsupervised sentence alignment algorithm (Bott, S. and Saggion, H., 2011) and then correcting it by a human editor to guarantee perfect alignment. In addition, original texts and their simplifications were automatically processed with our numerical expression recognition system so that we could observe how numerical expressions were transformed in the simplified document. The transformations observed on the Spanish corpus confirm the conclusions of studies

applied to the English language, where similar operations were indentified in the process of simplification. Concerning the size of the dataset in number of sentences, it comprises 570 sentences, 246 in the original set and 324 in the simplified set. Even though easy-to-read guidelines do not specifically concentrate on the treatment of numerical expressions, we have focused on the transformations of numerical information. Simplification operations applied to these expressions include: rounding, insertion of modifiers to account for the loss of precision, eliminating optional elements, etc. The transformations observed can be grouped into different kinds of operations: syntactic operations, lexical operations, content reduction and clarification. We have classified the operations over numerical expressions as follows:

1. Delete NumExp: the whole numerical expression is not preserved in the simplified sentence;
2. Delete sentence: the sentence containing the numerical expression is not preserved in the simplified document;
3. Keep NumExp: the numerical expression is maintained as such in the simplified sentence;
4. Change modifier + round: the modifier of the numerical expression is changed and the number is rounded;
5. Delete modifier + round: the modifier is deleted and the number itself is rounded;
6. Rewrite NumExp: the numerical expressions is re-written in letters; and
7. Delete modifier + keep number: the modifier is deleted keeping only the number.

Examples of each operation type are presented in Table 3.

[Here Table 3]

Table 4 shows how often the simplification of numerical expressions has been applied in the corpus (in percentages). We can see that over 50% of the numerical expressions are deleted in the Simplext corpus.

[Here Table 4]

The corpus analysis has led us to the conclusion that, given that the treatment of numerical expressions was not central to the simplification process, in many cases numerical information was lost, either through deleting the numerical expression itself or the entire sentence containing it. As a result, the simplified text does not contain all numerical information from the original. In spite of this, we can see interesting editing operations being applied, such as rounding and the addition or reformulation of modifiers. We note that the simplification guidelines used to produce the simplifications are not impositions and they leave quite a lot of freedom for the editors. These findings led us to undertake a second study to better understand how numerical information could be simplified.

3.4. Survey

In order to extend the set of possible operations used to simplify numerical expressions, we carried out a survey in addition to the analysis of the corpus. A subset of numerical expressions from the corpus has been selected in their original context (i.e. in the sentence they occur in the corpus). Spanish native speakers were asked to simplify freely the numerical information in such a way that the resulting expression be simpler and linguistically correct in the given context. The target types of numerical expressions used in the survey were: monetary expressions (18 million of Euros), percentages (54%), fractions (a quarter), physical measures (120,000 square kilometres) and general

quantities (3,000 boys). The objective of the survey was to extend the set of possible simplification operations observed in the corpus study. For this purpose, a questionnaire was developed using Google Forms. Experimental evaluation was carried out with 23 participants. All of them were Spanish native speakers with a university degree. There were 14 sentences in the questionnaire, with a total of 27 numerical expressions. Twelve of the original expressions had modifiers, while the remaining 15 did not.

For this sentence, we present a list of different simplified options of the numerical expressions marked in brackets, proposed by the participants. The sentence is: "[Alrededor de 390.000] personas han regresado a sus casas desde que vieran obligadas a desplazar se por las inundaciones causadas por las lluvias monzónicas de el pasado verano en Pakistán . ([Around 390,000 people] returned to their homes after they had been forced to leave due to floods caused by monsoon rains last summer in Pakistan.) ". For the numerical expression in the sentence "Alrededor de 390.000", the list of collected simplifications: "casi 400.000 (almost 400,000)", "unos 400.000 (some 400,000)", "alrededor de 400.000 (around 400,000)", "unas 400.000 (some 400,000)", "cerca de 400.000 (about 400000)", "casi 400.000 (almost 400,000)", "más de 300.000 (more than 400,000)", "casi 400.000 (almost 400,000)". We can see that several participants agree in the proposed simplifications. In Figure 2 we can see an example of a sentence from the survey.

[Here Figure 2]

Analysing the data collected in the survey we have observed the following operations applied by the participants to simplify the numerical expressions presented in each sentence:

1. Add modifier + round: the number is rounded and a modifier is added so that it is clear that the expression is not exact.

2. Change modifier + round: the number is rounded, and the original modifier is substituted for another one. The expression is not precise and the modifier is different from the original one.

3. Keep NumExp: the numerical expression is maintained as such in the original sentence.

4. Rewrite NumExp: the numerical expression is re-written in a different way, in letters or using other mathematical representation, etc.

5. Keep modifier + round: the number is rounded but the original modifier is kept, so that it is clear that the expression is not precise in spite of using the same modifier.

In Table 5 we show examples of operations applied by the subjects.

[Here Table 5]

Table 6 presents the operations observed in the simplification process in the survey and their frequency of use. We observe that operation "*Add modifier + round*" is very frequent and it was not present in the Simplext corpus. We consider this operation very important for simplification because it informs the reader about a possible rounding operation being applied. The three most common operations here are: *change modifier more round, keep numerical expression* and *rewrite numerical expression.*

[Here Table 6]

The conclusions drawn from the survey allow us to devise a simplification strategy to be included in our automatic simplification system. We identify three parts of a numerical expression: the modifier, the quantity, and the measurement unit. The modifier is treated where necessary, the quantity is rounded, and the unit, where present,

is kept without any change. The survey revealed that the most commonly used modifiers were “casi” (“almost”) and “más de” (“more than”). In spite of the observations from the corpus which indicate that numerical information is generally lost during simplification, we believe this is only due to the type of user involved in the Simplext project. We believe that instead of eliminating numerical information, a process by which the essential content is preserved, probably at the cost of loosing precision, could be beneficial, as the survey undertaken demonstrates.

3.5. A rule-base lexical component

Our simplification system is composed of the following components in the sequence described afterwards:

1. Text processing using FreeLing
2. Transformation of FreeLing output into XML representation
3. Application of grammars for numerical expression recognition
4. Simplification of target numerical expression
5. Sentence rewriting

These components have all been integrated in a plug-in developed in Java which can be used within the GATE system. FreeLing is used to carry out the basic analysis of the text: word and sentence recognition and parts-of-speech tagging. We also apply the FreelLing phrase structure parser but we do not use it for this work. A specialized module is used to produce XML annotations which are needed to transfer the linguistic information into GATE. In Figure 3 we can see the document analyzed by FreeLing and loaded in the GATE system. The highlighted annotation “16,4 millones” (“16.4 million”) has been tagged as a word with tag Zd.

[Figure 3]

The third component is the set of grammars described before, which were integrated into a named entity recognizer in GATE. This module produces annotations of type NumExp in the document which contains a feature that indicates the type of pre-modification of the numerical expression. The fourth component implements the simplification strategy which is most commonly observed in the survey: the quantity is always rounded and a set of rules is applied to the modifier chosen to account for loss of precision. In order to obtain the rounded number corresponding to the original quantity, auxiliary calculations using different methods of the package Math of Java are used. For example, if the original value of the quantity is “0.891”, we calculate its rounded value “1.0”. If measurement units are present in the original expression, they are also processed. The simplified version is made up of a selected modifier, the rounded numerical expression and optionally the units if these were present in the original text. In Table 7, we can see the selection of modifiers for the simplified numerical expression. If in the original expressions there is modifier, it is kept and the quantity is rounded. For the rest of the cases, the system compares the original quantity to rounded quantity and depending on the value, a modifier is selected. The modifiers chosen are the most commonly used ones in our survey. Since they are frequently used we claim they are most productive and easier to understand. The information to transform the expression is stored in features with appropriate values which are used in the rest of the pipeline.

[Here Table 7]

Finally, the last module rewrites the text, replacing the original numerical expression with the components added in the previous module. So, for each numerical expression, the feature with the simplified version is processed to replace it. After replacement,

post-processing of the text is carried out to correct any errors which arose from the treatment of the text by FreeLing, e.g. the transformation of contractions, such as “del” (“of the”) into their components “de + el” (“of the”).

As for linguistic accuracy of the output, the system was rather positively rated, with 83.56% (almost 84%) of the simplified sentences (all containing numerical expressions) considered correct; the meaning was also preserved reasonably well in the process of simplification. The corpus used for the evaluation had 57 texts with 73 sentences. One of the authors of the paper was in charge of rating the linguistic accuracy of the output. The qualitative analysis of the results revealed that most common errors that result in poor correction of the output sentence were bad treatment of comparative numerical expressions. In this case, we found errors in 4 different sentences where the treatment of comparative numerical expressions was bad. For example, for this original sentence “Las cifras de disoluciones se mantienen en 2010 similares a las de 2009, 22.435 frente a 21.875 , con un ligero incremento del 2,56 % .” (“Bankruptcy figures in 2010 are similar to those of 2009: 22,435 versus 21,875, a slight increase of 2.56%”), the output of the system was “Las cifras de disoluciones se mantienen en 2010 similares a las de 2009, más de 20000 frente a más de 20000, con un ligero incremento del casi 3%”. Here, we can see that the meaning preservation of the original sentence is lost. To measure the meaning preservation the evaluator compared the original and the simplified sentences using their own judgment.

4. Discussion

Our corpus analysis has shown that the most productive transformation is to eliminate either the original sentence or the numerical expression. However, this is too risky to

implement in an automatic system. We have observed that in the Simplext corpus there are over three numerical expressions per text and if these were to occur in different sentences, a lot of material would be excluded from the simplified version of the document. Our survey results suggest possible strategies for preserving the content, and this is why we suggest that numerical expression is simplified instead of eliminated, although slight loss of precision is inevitable. The latter is adjusted through the use of modifiers, and on the whole, the message of the original is more closely preserved in the simplified text. The aim of the survey was to gather simplification strategies used by humans, focusing on numerical expressions proper. This gave us a set of operations for the automatic simplification system to apply in the process of simplifying numerical expressions of original texts.

Our numerical expression simplification component was positively evaluated (both automatically for simplicity and with humans for grammaticality) as a part of a larger structure for automatic text simplification (Drndarevic B, Stajner S, Bott S, Bautista S and Saggion H., 2013). The aim of the evaluation is to test the degree of the simplification of the system and its components; and the grammaticality of the output and the preservation of meaning with respect to the original. To achieve the former, a set of Spanish readability formulae were used and we carried out evaluation with human annotators, who rated the degree of grammaticality and meaning preservation in a Likert-scale type of questionnaire. Two general conclusions can be made from the results obtained by the system in the evaluation of the complete simplification system: (1) both the syntactic simplification and the lexical transformations generally produce simpler output with respect to the original; (2) the combination of the two simplification processes generally produces a simpler output than either one individually. The system does not reach the simplification degree of manual transformations, but this is largely

due to the fact that summarisation and paraphrases are two most commonly applied techniques in the process of manual simplification, and as a result, a significant portion of the original content is eliminated.

It is important to note that using FreeLing we can identify and annotate different types and numerous cases of numerical expressions, whereas other tools based on machine learning, such as OpenNLP (OpenNLP), Maltparser (Nivre, 2003), Matetools (Bohnet, B. and Nivre, J., 2012), base their analysis on the corpus used to train them and use the Penn Treebank POS annotation, where only one annotation type is available for numerical expressions (CD). As a result, such tools cannot describe numerical expressions in more detail in the way FreeLing can.

Our approach was inspired by previous work on numerical simplification for English. However we did not apply the same approach blindly, and instead we performed a corpus analysis of simplified material and administered a simplification questionnaire to Spanish speakers in order to identify what strategies should be implemented in the automatic system. We found that similar strategies are applied to both languages, but we believe that before directly replicating an approach for another language, a careful study should be undertaken. Implementation techniques similar to those applied here (e.g. rule-based contextually aware procedures) could be used for languages close to Spanish.

One of the main problems we encountered is the treatment of comparative numerical expressions within a single sentence. For example, the original sentence “The monthly average in 2010 was 6,710 start-ups, as opposed to 6,635 in 2009” (“El promedio mensual en 2010 fue de 6.710 constituciones, frente a las 6.635 de 2009”) is simplified by our system into “The monthly average in 2010 was almost 7000 start-ups, as opposed to almost 7000 in 2009” (“El promedio mensual en 2010 fue de casi 7000

constituciones, frente a las casi 7000 de 2009."). This is due to the fact that our system applies the same simplification strategy to both numerical expressions without taking into account the context in which they occur, thus losing the sense of the original sentence. Such cases, where our current simplification approach fails, could be treated by further adjusting our JAPE grammars.

The presence of numerical information in a text impacts its readability. It is an important conclusion made in a study measuring the influence of different representations of numerical expressions on the reading process. In Rello et al. (2013) we studied the influence that different types of numerical expressions have on readability and understandability. An eye-tracking study with people with and without dyslexia was undertaken to measure the effect of rounded vs. exact numbers, digit vs. letters, and fractions vs. percentages. It was found that digits and percentages have a positive effect on readability and fractions have a positive effect on understandability for people with dyslexia.

Experimental psychology and cognitive neuropsychology have studied number processing and calculation over the last two decades. Many researchers have studied the cognitive processes that are responsible for number processing and calculation, with the aim of contributing to the improvement of teaching and learning processes. For example, Herrera and Macizo (2012), and Salguero and Alameda (2003), present findings that show that the frequency of use of a word or a number is an influential variable in the reading process. In addition, it seems that numerical expressions most frequently used require less recognition time. It is important to address in our future work the simplification of numerical expressions using the most frequently used ones.

5. Conclusions and future work

In this paper we have presented a numerical expression simplification component for Spanish to make numerical information more accessible. In order to build our component and with the aim of drawing conclusions about the kind of simplification operations that could be applied to numerical expressions, we have analyzed a parallel corpus of original and manually simplified news articles in Spanish to target different types of simplification operations. Simplifications have been applied by trained human editors, bearing in mind our target user, i.e. a person with cognitive disabilities, and following a series of easy-to-read guidelines. This kind of numerical simplifications would be useful for a more general public as well, such as older people, blind people, and deaf people. In addition, we have analyzed the answers collected from a survey in which human participants were asked to simplify numerical expressions from the corpus, in order to identify simplification strategies used by them. Our evaluation demonstrates that our components can identify complex numerical expressions with high precision and recall and that the simplifications obtained are correct in over 80% of the cases. Understandably, there are cases where our simplification component currently fails and its performance could be improved from various aspects.

Concerning the elimination of content, one possible approach could be to use automatic summarization techniques at the level of the numerical expressions (Drndarevic, B. and Saggion, H., 2012). A classifier could be developed which would enhance text simplification by helping decide which content to keep and what elements to delete, one of the features being the number of numerical expressions.

Out of the simplification strategies revealed in the survey, rewriting of a numerical expression or an entire sentence is not included in the current version of our component. We could try to tackle this through paraphrases, using a natural language generation system. As part of our future work, we intend to take syntactic context into

consideration when simplifying numerical expressions. It is important that the meaning of the sentences be preserved regardless of whether a part of the sentence is deleted or rewritten through the application of edit operations.

Although (Power, R. and Williams, S., 2012) presented an approach where less detailed numerical information is presented at the beginning of a document, gradually increasing its linguistic complexity as the text progresses, we have not implemented such a procedure here. On the one hand, the analysis of the corpus shows a regular treatment of numerical expressions irrespectively of their position in the text, and the one other, our survey analysis was carried out at the sentence level, ignoring factors such as sentence position or relation with other elements in the text.

References

Agencia Servimedia. (2010). Recuperado el 22 de 06 de 2013, de Agencia Servimedia: www.servimedia.es

Aluísio, S. M., Specia, L., Pardo, T.A., Maziero, E. and Fortes, R. (2008). Towards Brazilian Portuguese automatic text simplification systems. *ACM Symposium on Document Engineering 2008: 240-248.*

Anula, A. (2007). Tipos de Textos, Complejidad Lingüística y Facilitación Lectora. *Actas del Sexto Congreso de Hispanistas de Asia*, (pp. 45-61).

Aswani, N., Tablan,V., Bontcheva, K. and Cunningham, H. (2005). Indexing and Querying Linguistic Metadata and Document Content. *Proceedings of 5th International Conference on Recent Advances in Natural Language Processing.* Borovets, Bulgaria.

Bautista, S., Drndarevic, B., Hervás, R., Saggion, H. and Gervás, P. (2012). Análisis de la Simplificación de Expresiones Numéricas en Español mediante un estudio empírico. *Linguamática* .

Bautista, S., Gervás, P. and Madrid, R.I. (2009). Feasibility analysis for semiautomatic conversion of text to improve readability. *Proceedings of the Second International Conference on Information and Communication Technology and Accessibility.* Hammamet, Tunisia.

Bautista, S., Hervás, R., Gervás, P. Power, R. and Williams, S. (2011). How to Make Numerical Inormation Accessible: Experimental Identification of Simplification Strategies. *13th IFIP TC13 Conference on Human-Computer Interaction (INTERACT).* Lisbon, Portugal.

Bautista, S., Hervás, R., Gervás, P., Power, R. and Williams, S. (2013). A System for the Simplification of Numerical Expressions at Different Levels of Understandability. *Workshop Natural Language Processing for Improving Textual Accessibility.* Attlanta, USA.

Bisantz, A.M., Schinzing, S. and Munch, J. (2005). Displaying uncertainty: Investigating the effects of display format and specificity. *The Journal of the Human Factors and Ergonomics.*

Bohnet, B. and Nivre, J. (2012). A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. *ENMLP-CoNLL*, (pp. 1455-1465). Jeju Island, Korea.

Bott, S. and Saggion, H. (2011). An Unsupervised Alignment Algorithm for Text Simplification Corpus Construction. *Workshop on Monolingual Text-to-Text Generation.* Portland, USA.

Bott, S. and Saggion, H. (2012). Automatic Simplification of Spanish Text for e-Accessibility. *13th International Conference on Computers Helping People with Special Needs*, (págs. 54-56). Linz, Austria.

Bott, S., Rello, L., Drndarevic, B. and Saggion, H. (2012). Can Spanish Be Simpler? LexSiS: Lexical Simplification for Spanish. *The 24th International Conference on Computational Linguistics.* Mumbai, India.

Canning, Y. (2000). Cohesive simplification of newspaper text aphasic readers. *3rd Annual CLUK Doctoral Research Colloquium.*

Caseli, H.M., Pereira, T.F., Specia, L., Pardo, T.A.S, Gasperin, C. and Aluisio, S.M. (2009). Building a Brazilian Portuguese Parallel Corpus of Original and Simplified Texts. *In Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics.* Mexico City, Mexico.

Chandrasekar, R. and Srinivas, B. (1997). Automatic induction of rules for text simplification. *Knowledge-Based Systems* .

Chandrasekar, R., Doran, C. and Srinivas, B. (1996). Motivations and methods for text simplifications. *Proceedings of the 16th International Conference on Computational Linguistics*, (pp. 1041-1044). Copenhagen, Denmark.

Clark, D. (2010). *Young people reading and writing today: Whether, what and why.* National Literacy Trust.

Deeqa Jama, G. (2010). *Literacy: State of the nation.* National Literacy Trust.

Department of Education. (2003). *Skills for life.* United Kingdom.

Devlin, S and Tait, J. (1998). The use of a Psycholinguistic database in the Simplification of Text for Aphasic Readers. *Linguistic Databases* , 161-173.

Devlin, S. and Unthank, G. (2006). Helping aphasic people process online information. *Proceedings of the 8th international ACM SIGACCESS conference on Computers and Accessibility* (pp. 225-226). New York, USA.: ACM.

Dieckmann, N., Slovic, P. and Peters, E. (2009). The use of narrative evidence and explicit likelihood by decision markers varying in numeracy. *Risk Analysis* .

Drndarevic B, Stajner S, Bott S, Bautista S and Saggion H. (2013). Automatic Text Simplification in Spanish: A Comparative Evaluation of Complementing Modules. *International Conference on Intelligent Text Processing and Computational Linguistics.* Samos, Greece.

Drndarevic, B. and Saggion, H. (2012). Reducing Text Complexity through Automatic Lexical Simplification: an Empirical Study for Spanish. *Sociedad Española para el Procesamiento del Lenguaje Natural* , 13-20.

Drndarevic, B., Saggion, H. (2012). Towards Automatic Lexical Simplification in Spanish: An Empirical Study. *Workshop Predicting and Improving Text Readability for Target Reader Populations.* Montreal, Canada.

EAGLES. (n.d.). Retrieved 06 22, 2013, from EAGLES: http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html

Freyhoff, G., Hess, G., Kerr, L, Menzel, E., Tronbacke, B. and Veken, K.V.D. (1998). *European guidelines for the production of easy-to-read information*. Retrieved 06 22, 2013, from http://www.osmhi.org/contentpics/139/European Guidelines for ETR publications.pdf

Grice, H. (1975). Logic and Conversation. *Syntax and Semantics* , 41-58.

Herrera, A. and Macizo, P. (2012). Cómo leemos los números? (How we read numbers?). *Ciencia Cognitiva* , 44-47.

Klevanov, B.B., Knight, K. and Marcu, D. (2004). Text simplification for information-seeking applications. *On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science* (pp. 735-747). Springer Verlag.

Krifka, M. (2002). Be brief and vague! And how bidirectional optimality theory allows for Verbosity and Precision. *Sounds and Systems: Studies in Structure and Change: A Festschrift for Theo Vennemann*, (pp. 439-458). Berlin.

Max, A. (2006). Writing for language-impaired readers. *Proceeding on International Conference on Intelligent Text Processing and Computational Linguistics*, (pp. 567-570). Mexico City, Mexico.

Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K. and Wilks, Y. (2002). Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering- Special Issue on Robust Methods in Analysis of Natural Language Data* , 257-274.

Mishra, H., Mirshra, A. and Shiy B. (2011). In praise of vagueness: malleability of vague information as a performance booster. *Psychological Science* .

Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. *Proceedings of the 8th International Workshop on Parsing Technologies*, (pp. 149-160). Nancy, France.

OpenNLP. (n.d.). Retrieved 06 22, 2013, from OpenNLP: http://opennlp.apache.org/documentation.html

Padró, Ll., Collado, M., Reese, S., loberes, M. and Castelln, I. (2010). Freeling 2.1: Five years of open-source language processing tools. *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

Peters, E., Hibbard, J., Slovic, P. and Dieckmann, N. (2007). Numeracy skill and the communication, comprehension, and use of risk-benefit information. *Health Affairs* , 741-748.

Petersen, S.E. and Ostendorf, M. (2007). Text simplification for language learners: a corpus analysis. *Proceedings of Workshop on Speech and Language Technology for Education.*

Power, R. and Williams, S. (2012). Generating numerical approximations. *Computational Linguistics* .

Proyecto Simplext. (2010). Retrieved 06 22, 2013, from Proyecto Simplext: www.simplext.es

Rello, L., Bautista, S., Baeza-Yates, R., Gervás, P., Hervás, R. and Saggion, H. (2013). One half or 50%? An eye-tracking study of number representation readability. *14th IFIP TC13 Conference on Human-Computer Interaction.* Cape Town, South Africa.

Saggion, H., Gómez-Martínez, E., Esteban Etayo, E., Anula, A. and Bourg, L. (2011). Text Simplification in Simplext. Making Text More Accessible. *Procesamiento del Lenguaje Natural , 47*, 341-342.

Salguero, M. and Alameda, J. (2003). El procesamiento de los números y sus implicaciones educativas (Number processing and its educational implications). *XXI Revista de Educación (Educational Journal)* , 181-189.

Siddharthan, A. (2002). Resolving attachment and clause boundary amgiguities for simplifying relative clause constructs. *Proceedings of the Student Research Workshop, 40th Meeting of the Association of Computational Linguistics.*

Siddharthan, A. (2003). Syntactic Simplification and Text Cohesion. *Ph.D dissertation, research and language and computation.*

Snow, C. (2002). *Reading for understanding: toward and R&D program in reading comprehension.* Santa Monica, CA. United States: Science and T.P.I.R. Corporation.

Specia, L, Jahuar, S.K. and Milhacea, R. (2012). SemEval-2012 Task 1: Englosh Lexical Simplification. *Proceedings of the SemEval Conference.* Montréal, Canada.

Specia, L. (2010). Translating from Complex to Simplified Sentences. *Proceedings of Computational Processing of the Portuguese Language.* Porto Alegre, RS, Brazil.

The Plain Language Action and Information Network (PLAIN). (2005). Retrieved 06 22, 2013, from Plain Language: http://www.plainlanguage.gov

W3C. (2008). *Web Content Accessibility Guidelines*. Retrieved 06 22, 2013, from http://www.w3.org/TR/WCAG20/

Williams, S. and Power, R. (2009). Precision and mathematical form in first and subsequent mentions of numerical facts and their relation to document structure. *12th European Workshop on Natural Language Generations.* Athens, Greece.

Williams, S., Reiter, E. and Osman, L.M. (2003). Experiments with discourse-level choices and readability. *Proceedings of the European Natural Language Generation Workshop*, (pp. 127-134). Budapest, Hungary.

Zhu, Z., Bernhard, D. and Gurevych, I. (2010). A monolingual tree-based translation model for sentence simplification. *Proceedings of the 23rd International Conference on Computational Linguistics.* Beijing, China.

Appendix

Table 1

Operation	# quantity
Sentences	1149
Tokens	40120
Avg Sentences per Document	5,745
Avg Tokens per Sentence	34,92

Figure 1

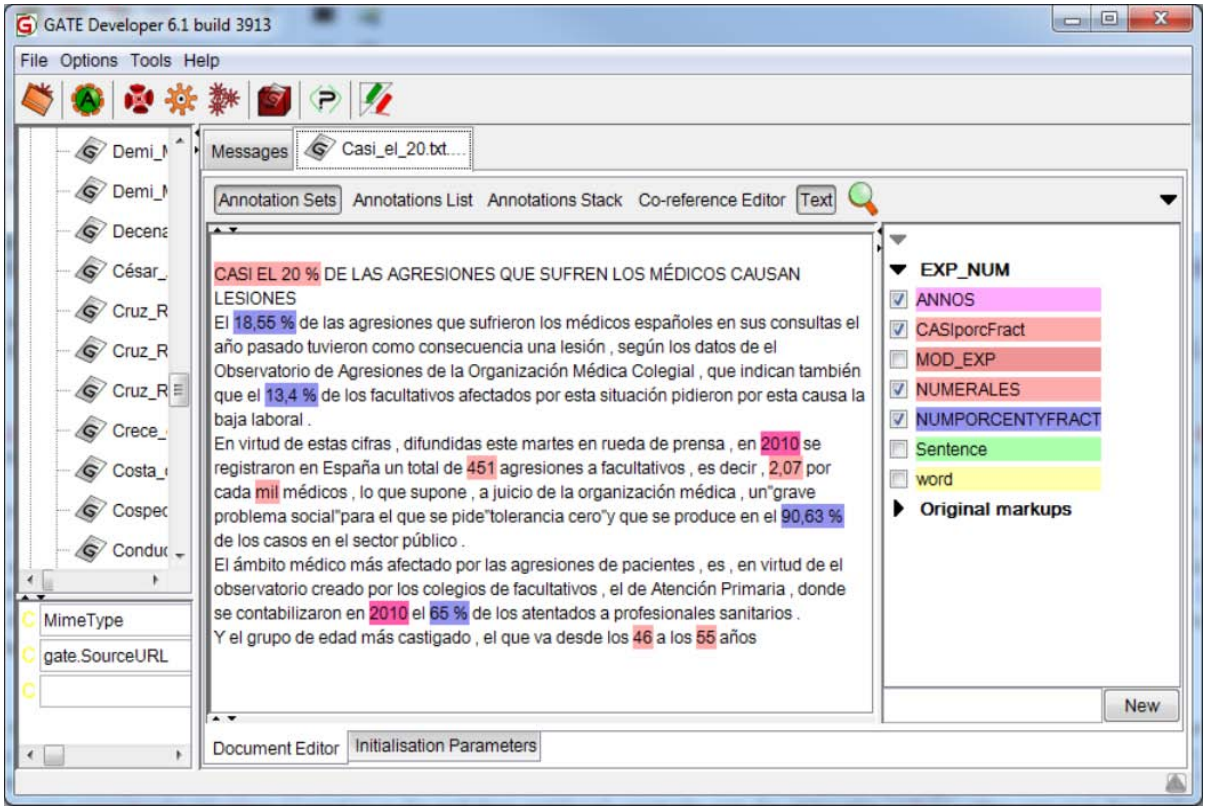

Table 2

ANNOTATION	NumExp	Example
CASIporcFract	casi + Zp (almost + Zp)	casi un cuarto (almost a quarter)
DURANTENUM	durante + Z (for + Z)	durante 24 días (for 24 days)

MASDENUM	más de + Z (more than + Z)	más de 50.000 (more than 50.000)
MASDEPART	más de + Zd (more than +Zd)	más de 30 millones (more than 30 million)
MASDEporcFract	más de + Zp (more than + Zp)	más del 40% (more than 40%)
NUMERALES	Z	34.589
NUMMAGNITUDES	Zu	32 metros (32 meters)
NUMMONETARIAS	Zm	1.400 euros
NUMPARTITIVO	Zd	32 millones (32 million)
NUMPORCENTYFRACT	Zp	75%
UNASMagnit	unas + Zu	unas 700 millas
UNASNUM	unas + Z (some + Z)	unas 20.000 (some 20.000)
MOD_EXP	modifier	alrededor, menos de… (around, less than, ...)

Table 3

Operation	Original example	Simplified example
Delete NumExp	In 2010 Amnesty International registered cases of torture and other forms of abuse in at least 111 countries, unfair trials in 55, infringements on freedom of speech in 96 and arresting prisoners of conscience in 48 countries. (Amnistía Internacional ha	This organisation proved the existence of cases of torture, abuse, and unfair trials in 2010. (Esta organización ha probado casos de tortura,

	documentado durante 2010 casos de tortura y otros malos tratos en al menos 111 países, juicios injustos en 55, restricciones de libertad de expresión en 96 y presos de conciencia encarcelados en 48.)	malos tratos y juicios injustos en 2010.)
Delete sentence	The box office was similarly affected, experiencing a decline from 68.3 million Euros to 65 million. (La taquilla se resintió de forma similar, y bajó de 68,3 millones de euros a 65 millones.)	Sentence deleted (oración eliminada)
Keep NumExp	6,300 light years (6.300 años luz)	6,300 light years (6.300 años luz)
Change modifier + round	Almost 7,400 million Euros (Casi 7.400 millones de euros)	More than 7,000 million Euros (Más de 7.000 millones de euros)
Delete modifier + round	Around 1,9 million houses (Unos 1.9 millones de casas)	2 million houses (2 millones de casas)
Rewrite NumExp	More than 540,000 people (Más de 540.000 personas)	Half a million people (Medio millón de personas)
Delete modifier +	Around 300,000 children (Unos 300.000 niños)	300,000 children (300.000 niños)

keep number		

Figure 2

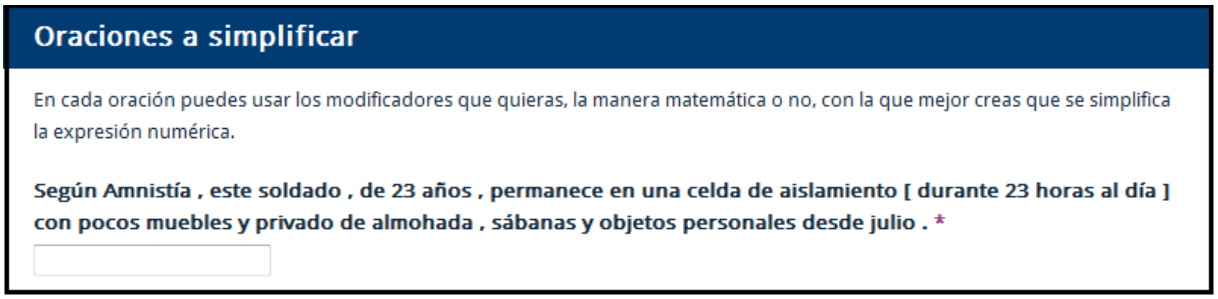

Table 4

Simplification Operation	% Use
Delete NumExp	44.4%
Delete Sentence	25.9%
Keep NumExp	7.4%
Change modifier + round	7.4%
Delete modifier + round	7.4%
Rewrite NumExp	3.7%
Delete modifier + keep number	3.7%

Table 5

Operation	Original example	Simplified example
Add modifier + round	48	Almost 50 (Casi 50)
Change modifier + round	At least 111 countries (Al menos 111 países)	More than 100 countries (Más de 100 países)
Keep NumExp	3,000 children (3.000 niños)	3,000 children (3.000 niños)

Rewrite NumExp	26% 23 hours a day (23 horas al día)	A fourth (un quarto) Almost all day (Casi todo el día)
Keep modifier + round	Around 5,400 million Euros (Casi 5.400 millones de Euros)	Around 5,000 million Euros (Casi 5.000 millones de Euros)

Table 6

Simplification Operation	% Use
Add modifier + round	33.3%
Change modifier + round	22.2%
Keep NumExp	18.5%
Rewrite NumExp	18.5%
Keep modifier + round	7.4%

Figure 3

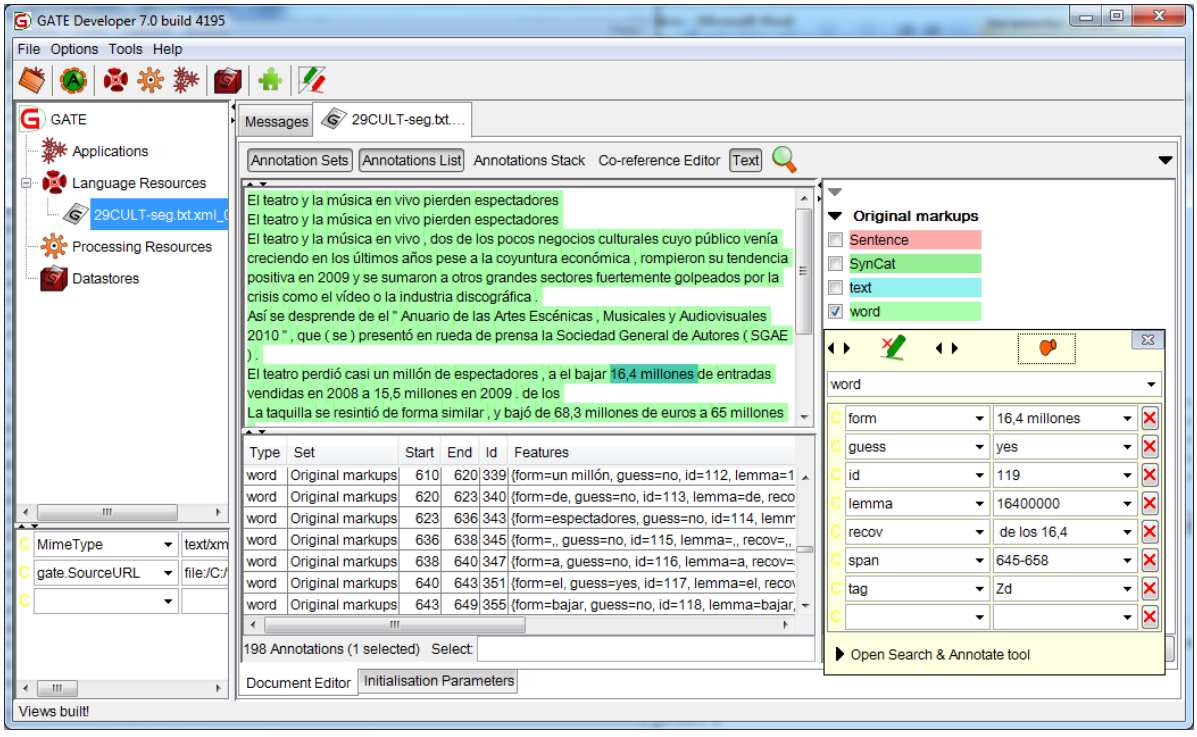

Table 7

	Simplified Numerical Expression	
Cases	Modifier	Quantity
If modifier exists in the original expression, then it is kept. "Alrededor de 3952 millones (Around 3952 million)"	Original modifier is kept Alrededor de (Around)	Original quantity is rounded 4000 millones (4000 million)
If modifier does not exit, and originalQuantity > roundedQuantity 26.7% > 25%	Add modifier Más de (More than)	Original quantity is rounded 25% (25%)
If modifier does not exit, and originalQuantity < roundedQuantity 487 < 500	Add modifier Casi (Almost)	Original quantity is rounded 500 (500)
If modifier does not exit, and originalQuantity = roundedQuantity 20000 = 20000	Add modifier Unos (Some)	Original quantity is rounded 20000 (20000)