# A system for the simplification of numerical expressions at different levels of understandability

# A System for the Simplification of Numerical Expressions at Different Levels of Understandability

**Susana Bautista, Raquel Hervás, Pablo Gervás**
Universidad Complutense de Madrid
Prof. José García Santesmases
Madrid, Spain
{subautis,raquelhb}@fdi.ucm.es
pgervas@sip.ucm.es

**Richard Power, Sandra Williams**
Department of Computing,
The Open University
Milton Keynes,
MK76AA, UK
r.power@open.ac.uk
s.h.williams@open.ac.uk

## Abstract

The purpose of this paper is to motivate and describe a system that simplifies numerical expression in texts, along with an evaluation study in which experts in numeracy and literacy assessed the outputs of this system. We have worked with a collection of newspaper articles with a significant number of numerical expressions. The results are discussed in comparison to conclusions obtained from a prior empirical survey.

## 1 Introduction

A surprisingly large number of people have limited access to information because of poor literacy. The most recent surveys of literacy in the United Kingdom reveal that 7 million adults in England cannot locate the reference page for plumbers if given the Yellow Pages alphabetical index. This means that one in five adults has less literacy than the expected literacy in an 11-year-old child (Jama and Dugdale, 2010; Williams et al., 2003a; Christina and Jonathan, 2010). Additionally, almost 24 million adults in the U.K. have insufficient numeracy skills to perform simple everyday tasks such as paying household bills and understanding wage slips. They would be unable to achieve grade C in the GCSE maths examination for 16-year-old school children (Williams et al., 2003a).

"The Standard Rules on the Equalization of Opportunities for Persons with Disabilities" by United Nations (1994) state that all public information services and documents should be accessible in such a way that they could be easily understood. If we focus on numerical information, nowadays, a large percentage of information expressed in daily news or reports comes in the form of numerical expressions (economic statistics, demography data, etc) but many people have problems understanding the more complex expressions. In the text simplification process, different tasks are carried out: replacing difficult words, splitting sentences, etc., and the simplification of numerical expressions is one of them.

A possible approach to solve this important social problem of making numerical information accessible is to rewrite difficult numerical expressions using alternative wordings that are easier to understand. For example, the original sentence, "25.9% scored A grades" could be rewritten by "Around 26% scored A grades". In our study we define a "numerical expression" as a phrase that presents a quantity, sometimes modified by a numerical hedge as in these examples: 'less than a quarter' or 'about 98%'. Such an approach would require a set of rewriting strategies yielding expressions that are linguistically correct, easier to understand than the original, and as close as possible to the original meaning. Some loss of precision could have positive advantages for numerate people as well as less numerate. In rewriting, hedges play also an important role. For example, '50.9%' could be rewritten as 'about a half' using the hedge 'about'. In this kind of simplification, hedges indicate that the original number has been approximated and, in some cases, also the direction of the approximation.

This paper presents a system developed for automated simplification of numerical expressions. Experts in simplification tasks are asked to validate the



*Proceedings of the 2th Workshop of Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, pages 10–19,
Atlanta, USA, 14 June 2013. 

simplifications done automatically. The system is evaluated and the results are discussed against conclusions obtained from previous empirical survey.

## 2 Previous work

Text simplification, a relative new task in Natural Language Processing, has been directed mainly at syntactic constructions and lexical choices that some readers find difficult, such as long sentences, passives, coordinate and subordinate clauses, abstract words, low frequency words, and abbreviations.

The rule-based paradigm has been used in the implementation of some systems for text simplification, each one focusing on a variety of readers (with poor literacy, aphasia, etc) (Chandrasekar et al., 1996; Siddharthan, 2003; Jr. et al., 2009; Bautista et al., 2009).

The transformation of texts into easy-to-read versions can also be phrased as a translation problem between two different subsets of language: the original and the easy-to-read version. Corpus-based systems can learn from corpora the simplification operations and also the required degree of simplification for a given task (Daelemans et al., 2004; Petersen and Ostendorf, 2007; Gasperin et al., 2009).

A variety of simplification techniques have been used, substituting common words for uncommon words (Devlin and Tait, 1998), activating passive sentences and resolving references (Canning, 2000), reducing multiple-clause sentences to single-clause sentences (Chandrasekar and Srinivas, 1997; Canning, 2000; Siddharthan, 2002) and making appropriate choices at the discourse level (Williams et al., 2003b). Khan et at. (2008) studied the tradeoff between brevity and clarity in the context of generating referring expressions. Other researchers have focused on the generation of readable texts for readers with low basic skills (Williams and Reiter, 2005), and for teaching foreign languages (Petersen and Ostendorf, 2007).

Previous work on numerical expressions has studied the treatment of numerical information in different areas like health (Peters et al., 2007), forecast (Dieckmann et al., 2009), representation of probabilistic information (Bisantz et al., 2005) or vague information (Mishra et al., 2011). In the NUMGEN project (Williams and Power, 2009), a corpus of numerical expressions was collected and a formal model for planning specifications for proportions (numbers between 0 and 1) was developed. The underlying theory and the design of the working program are described in (Power and Williams, 2012).

## 3 Experimental identification of simplification strategies for numerical information

In order to analyze different simplification strategies for numerical expressions, first we have to study the mathematical complexity of the expressions. Expressions can be classified and a level of difficulty can be assigned. A study about the simplification strategies selected by experts to simplify numerical expressions expressed as decimal percentages in a corpus was carried out in Bautista et al. (2011b). Other important aspect of the simplification task is the use of hedges to simplify numerical expressions in the text. A study was performed in Bautista et al. (2011a) to analyze the use of hedges in the simplification process. This study was done with experts in simplification tasks. A set of sentences with numerical expressions were presented and they had to rewrite the numerical expressions following some rules. Several hypotheses were expressed and analyzed to understand experts' preferences on simplification strategies and use of hedges to simplify numerical expressions in the text. The main conclusions from the study were:

**Conclusion 1:** When experts choose expressions for readers with low numeracy, they tend to prefer round or common values to precise values. For example, halves, thirds and quarters are usually preferred to eighths or similar, and expressions like *N in 10* or *N in 100* are chosen instead of *N in 36*.

**Conclusion 2:** The value of the original proportion influences the choice of simplification strategies (fractions, ratios, percentages). With values in the central range (say 0.2 to 0.8 in a 0.0 to 1.0 scale) and values at the extreme ranges (say 0.0-0.2 and 0.8-1.0) favoring different strategies.

**Conclusion 3:** When writers choose numerical expressions for readers with low numeracy, they only use hedges if they are losing precision.

## 4 A system for adapting numerical expressions

In this first prototype, only numerical expressions defined as percentages are adapted. From an input text, the percentage numerical expressions are detected, a target level of difficulty is chosen and the simplified version of the text is generated by replacing the original numerical expression with the adapted expression.

### 4.1 Numerical expression

A numerical expression consists of: (1) a numerical value, a quantity which may be expressed with digits or with words; (2) an optional unit accompanying the quantity (euro, miles, ...); and (3) an optional numerical hedge modifier (around, less than, ...). Some examples of numerical expressions used in our experiments are: 'more than a quarter', 'around 98.2%', 'just over 25 per cent' or 'less than 100 kilometres'.

### 4.2 Levels of difficulty

The Mathematics Curriculum of the Qualifications and Curriculum Authority (1999) describes a number of teaching levels and we assume that concepts to be taught at lower levels will be simpler than ones taught at higher levels. Following this idea a Scale of Mathematic Concepts is defined to identify the different levels of difficulty to understand mathematic concepts. The scale defined from less to greater difficulty is: numerical expression in numbers (*600*), words (*six*), fractions (*1/4*), ratios (*1 in 4*), percentages (*25%*) and decimal percentages (*33.8%*).

From the Scale of Mathematic Concepts defined, different levels of difficulty are considered in our system. There are three different levels (from easiest to hardest):

1. *Fractions Level*: each percentage in the text is adapted using fractions as mathematical form for the quantity, and sometimes a hedge is used.

2. *Percentages without decimals Level (PWD)*: the system rounds the original percentage with decimals and uses hedges if they are needed.

3. *Percentages with decimals Level*: This is the most difficult level where no adaptation is performed.

The system operates only on numerical expressions at the highest levels of the scale (the most difficult levels), that is, numerical expression given in percentages or decimal percentages, adapting them to other levels of less difficulty. So, the user can select the level to which adapt the original numerical expression from the text. Using the interface of the system, the level of difficulty is chosen by the final user and the numerical expressions from the text with higher level of difficulty than the level chosen are adapted following the rules defined.

### 4.3 Set of strategies

A set of strategies is defined so they can be applied to adapt the original numerical expression. The quantity of the expression is replaced with another expression and sometimes numerical hedges are added to create the simplified numerical expression.

The use of hedges to simplify numerical expression can be influenced by three parameters. The first is the type of simplification depending on the mathematical knowledge of the final user. The second is the simplification strategy for the choice of the final mathematical form. And the last is the loss of precision that occurs when the expression is simplified.

Out of the European Guidelines for the Production of Easy-to-Read Information for People with Learning Disability (Freyhoff et al., 1998), only one involves the treatment of numbers: "Be careful with numbers. If you use small numbers, always use the number and not the word". For example, if the texts says 'four', the system adapts it by '4' following this European Guideline. This strategy is applied by the system at all levels.

There are other strategies to adapt numerical expressions in the form of percentage to other levels of difficulty: (1) replace decimal percentages with percentages without decimals; (2) replace decimal percentages with ratios; (3) replace percentages with ratios; (4) replace decimal percentages with fractions; (5) replace percentages with fractions; (6) replace ratios with fractions; (7) replace numerical expressions in words with numerical expressions in digits.

At each level of difficulty, a subset of the strategies is applied to simplify the numerical expression. For the *Fractions Level* the strategies 4, 5 and 7 are used. For the *Percentages with decimals Level* the strategies 1 and 7 are applied. And for the last

level, *Percentages without decimals Level* only the last strategy, number 7, is used.

### 4.4 System operation

The system takes as input the original text. The user of the system has to choose the level of difficulty. A set of numerical expressions are selected and a set of transformations is applied to adapt them, generating as output of the system a text with the numerical expressions simplified at the chosen level.

The system works through several phases to adapt the numerical expressions in the input text. Some of them are internal working phases (2, 4 and 5). The rest of them (1, 3 and 6) are phases where the user of the system plays a role. The phases considered in the system are:

1. **Input text**: an original text is selected to adapt its numerical expressions.

2. **Mark Numerical Expressions**: the numerical expressions that can be adapted are marked.

3. **Choose the level of difficulty**: the user chooses the desired level of difficulty for the numerical expressions in the text.

4. **Adapt the numerical expression from the text**: each numerical expression is adapted if the level of the numerical expression is higher than the level of difficulty chosen.

5. **Replace numerical expression in the text**: adapted numerical expressions replace the originals in the text.

6. **Output text**: the final adapted version of the text is presented to the user.

The next subsections presents how the system acts in each phase and what kind of tools are used to achieve the final text.

#### 4.4.1 Phase 1: Input text

In this first phase, a plain text is chosen as input to the system to adapt its numerical expressions. Using a Graphical User Interface (GUI) in Java, the user can upload an original text.

#### 4.4.2 Phase 2: Mark numerical expressions

For the text chosen, the system executes the *Numerical Expression Parser*[1]. Using this parser the numerical quantities are annotated with their type (cardinal, fraction, percentage, decimal percentage, etc.), their format (words, digits), their value *(Vg)*, their units, and hedging phrases, such as 'more than'. The input to the program is the plain text file and the output is the text with sentences and numerical expressions annotated in XML format. In the following code we can see how a numerical quantity is annotated in the parser.

> Overall figures showed the national pass rate soared
> <**numex hedge="above" hedgesem="greaterthan" type="percentage" format="digits" Vg="0.97"**>
> above 97% <**/numex**>

The XML file is treated by the system and numerical expressions are marked in the original text. So, the user can see which numerical expressions are going to be adapted by the system (in the next phase) depending on the level of difficulty chosen.

#### 4.4.3 Phase 3: Choose the level of difficulty

The user of the system chooses the level of difficulty to adapt the original numerical expressions. There are three levels: *fractions*, *percentages without decimals* and *percentages with decimals*.

#### 4.4.4 Phase 4: Adapt the Numerical Expressions

After deciding the level of difficulty, the system has to adapt each numerical expression to generate the final version. The process of simplification has two stages: obtaining the candidate and applying the adaptation and hedge choice rules.

From the XML file produced by the parser the following information for a numerical expression is obtained: (1) if there is or not hedge and the kind of hedge; (2) the type (cardinal, fraction, percentage, decimal percentage) and format (digits or words) of the original numerical expression; (3) the *given value (Vg)* translated from the original numerical expression value of the text; and (4) the units from the

[1]For more details see (Williams, 2010)

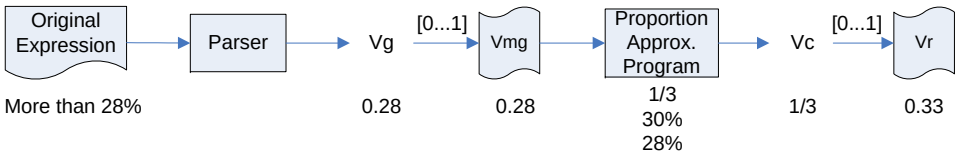


Figure 1: Obtaining the candidate for simplification. The original expression is annotated by the parser (Vg), and this value is normalized (Vmg). A candidate substitute value (Vc) is chosen from the *proportion approximation program* and normalized (Vr).

original expression (M, ins, grams). For example, if in the text the original numerical expression is a percentage like '25.9%', there is no hedge, the type is 'decimal percentage', the format is 'digits', *Vg* is 0.259 and there are no units. In the expression, '20 grams', there is no hedge, the type is 'cardinal', the format is 'digits', *Vg* is 20 and the parser annotates the units with 'g'.

The given value *Vg* annotated by the parser is transformed into a value between 0 to 1, referred to as *mapping given value (Vmg)*, which represents the proportion under consideration. This value is given as input to the *proportion approximation program* (Power and Williams, 2012), which returns a list of candidates for substitution. From this list, the first option is taken as *candidate substitute value (Vc)*, because the program returns them in decreasing order of precision. This means that the most precise candidate at the required level of difficulty is chosen. The program also might return the values "none" and "all" if the input value is close to 0 or 1, respectively. From the *Vc* we calculate the *rounded value (Vr)* corresponding to the normalization of the candidate value between 0 to 1. For example, if *Fraction level* is chosen, for the original expression "more than 28%" with *Vmg=0.28*, the system chooses *Vc*=1/3 with *Vr=0.33*. The whole process can be seen in Figure 1.

An additional level of adaptation is required beyond simple replacement with the candidate substitute value. If the original numerical expressions in the text are difficult to understand, the system must adapt them to the desired level of difficulty. For each numerical expression, the system only applies the adaptation rules if the difficulty level of the numerical expression is higher than the level of difficulty chosen by the user. This is captured by a set of three adaptation rules:

- If the type of the numerical expression is 'cardinal' and the format is 'words' then the candidate to be used in the simplification is *Vg*. For example, if the original numerical expression is 'six', it will be replaced by '6'.

- In a similar way, if the type is 'fraction' (the lowest possible level of difficulty) and the format is also 'words' then the candidate is obtained by applying the *proportion approximation program*. For example, if the original numerical expression is 'a quarter', it would be replaced by '1/4'.

- If the type is 'percentages' or 'decimal percentages' and the format is 'digits' then the candidate is calculated by the *proportion approximation program* provided that the level of difficulty chosen in the GUI was lower than the level of the calculated numerical expression.

In order to complete the simplification, the system has to decide if a hedge should be used to achieve the final version of the adapted numerical expression. This decision is taken based on the difference in value between the value of the original expression in the text *(Vg)* and the value of the candidate substitute *(Vc)* (as given by the relative difference between the normalized values *Vr* and *Vmg* calculated in the first stage). The actual hedge used in the original expression (if any) is also considered. The various possible combinations of these values, and the corresponding choice of final hedge, are described in Table 1, which presents all possible options to decide in each case, the hedge and the value corresponding to the final numerical expression. For example, if the original expression is "more than 28%", we have Vc=1/3, Vmg=0.28 and Vr=0.33. Then Vr>Vmg so the corresponding choice of the final hedge is in the

| OriginalNumExp | if Vr>Vmg | if Vr=Vmg | if Vr<Vmg |
|---|---|---|---|
| **more than OrigValue** | around Vc | more than Vc | more than Vc |
| **exactly OrigValue** | less than Vc | exactly Vc | more than Vc |
| **less than OrigValue** | less than Vc | less than Vc | around Vc |
| **OrigValue** | around Vc | Vc | around Vc |

Table 1: Hedge Choice Rules. For each original expression (OrigValue), the normalized values *(Vmg, Vr)* are used to determinate the hedge chosen for the simplified expression. The final version is composed by the hedge chosen and the candidate value *(Vc)*

first column of Table 1 ("around") and the simplified expression is "around 1/3".

When the user chooses the *Fraction Level* in the system, every numerical expression with difficulty level greater than fraction level will be replaced by a numerical expression expressed in fraction form. Depending on the values *Vr* and *Vmg*, the appropriate hedge will be chosen.

#### 4.4.5 Phase 5: Replace numerical expressions

Once the system has applied its rules, an adapted version is available for each original numerical expression which was more difficult than the target difficulty level. The output text is obtained by replacing these difficult expressions with the corresponding simplified version.

## 5 Evaluation of the system

This section presents the evaluation of the system, describing the materials, experiment, participants and results of the evaluation.

### 5.1 Materials

We selected for the experiment a set of eight candidate sentences from the NUMGEN corpus, but the number of numerical expressions was larger as some sentences contained more than one proportion expression. In total we had 13 numerical expressions. We selected sentences with as many variations in context, precision and different wordings as possible. The range of proportions values was from points nearly 0.0 to almost 1.0, to give coverage to a wide spread of proportion values. We considered values in the central range (say 0.2 to 0.8) and values at the extreme ranges (say 0.0-0.2 and 0.8-1.0). We also classified as common values the well-known percentages and fractions like 25%, 50%, 1/4 and 1/2, and as uncommon values the rest like 15% or 6/7.

### 5.2 Experiment

To evaluate the system a questionnaire was presented to a set of human evaluators. The experiment was created and presented on SurveyMonkey[2], a commonly-used provider of web surveys. For each original sentence, we presented two possible simplifications generated by the system. Participants were asked to use their judgement to decide whether they agreed that the simplified sentences were acceptable for the original sentence. A Likert scale of four values (Strongly Disagree, Disagree, Agree, Strongly Agree) was used to collect the answers.

In the survey only two levels of adaptation from the original sentence were presented. The first option generated by the system was for the *Fractions level*. The second option generated by the system was for the *Percentages without decimals (PWD)*.

### 5.3 Participants

The task of simplifying numerical expressions is difficult, so we selected a group of 34 experts made up of primary or secondary school mathematics teachers or adult basic numeracy tutors, all native English speakers. This group is well qualified to tackle the task since they are highly numerate and accustomed to talking to people who do not understand mathematical concepts very well. We found participants through personal contacts and posts to Internet forums for mathematics teachers and numeracy tutors.

### 5.4 Results

The answers from the participants were evaluated. In total we collected 377 responses, 191 responses for the *Fraction level* and 186 responses for the *Percentage without decimals (PWD)*. Table 2 shows the average from the collected responses, considering 1

[2]http://www.surveymonkey.com/s/WJ69L86

| Level | Total average | Values | Average | Values | Average |
|---|---|---|---|---|---|
| **Fraction** | 2,44 | Central | 2,87 | Common | 2,59 |
| | | Extreme | 2,14 | Uncommon | 1,21 |
| **PWD** | 2,96 | Central | 3,00 | Common | 2,80 |
| | | Extreme | 2,96 | Uncommon | 3,22 |

Table 2: System Evaluation: Fraction Level and Percentages Without Decimals (PWD)

| Opinion | Fraction Level | PWD Level |
|---|---|---|
| **Strongly Disagree** | 19% | 6% |
| **Disagree** | 27% | 15% |
| **Agree** | 43% | 56% |
| **Strongly Agree** | 11% | 23% |

Table 3: Opinion of the experts in percentages

to 4 for strongly disagree to strongly agree. In addition, Table 3 shows the distribution in percentages of the opinion of the experts. At the *Fraction level*, there is not too much difference between the average of the answers of the experts that agree with the system and those that disagree. Most experts are neutral. But for the *PWD level* the average shows that most experts agree with the simplification done.

We have also analyzed the answers considering two different criteria from the original numerical expressions: when they are central (20% to 80%) or extreme values (0% to 20% and 80% to 100%), and when the original numerical expressions are common or uncommon values. In general terms, the experts think that the simplification done by the system in the *PWD level* is better than the simplification done in the *Fraction level*. They disagree specially with the simplification using fractions in two cases. One is the treatment of the extreme values where the system obtains as possible candidates "none" and "all"[3]. Another case is when uncommon fractions are used to simplify the numerical expression, like for example 9/10. In these two cases the average is lower than the rest of the average achieved.

### 5.5 Discussion

The system combines syntactic transformations (via the introduction of hedges) and lexical substitutions (by replacing actual values with substitution candidates and transforming quantities expressed as words into digits) to simplify the original numerical expression. These kinds of transformations are different from those used by other systems, which rely only on syntactic transformations or only on lexical substitutions. Rules are purpose-specific and focused on numerical expressions. With this kind of transformations the readability of the text improves in spite of the fact that the resulting syntactic structure of the numerical expression is more complicated, due to the possible presence of hedges. For example, for a original numerical expression like '25.9%' the system generates the simplified 'more than a quarter' which is easier to understand even though longer and syntactically more complex.

With respect to coverage of different types of numerical expressions, this system does not consider *ratios* as a possible simplification strategy because the *proportion approximation program* does not use them as candidates to simplify a proportion. This possibility should be explored in the future.

Another observation is that the system does not consider the context of the sentence in which the numerical expression occurs. For example, if the sentence makes a comparison between two numerical expressions that the system rounded to the same value, the original meaning is lost. One example of this case is the following sentence from the corpus: "One in four children were awarded A grades (25.9%, up from 25.3% last year)". Both percentages '25.9%' and '25.3%' are simplified by the system using 'around 1/4' and the meaning of the sentence is lost. Thus we should consider the role of context (the set of numerical expressions in a given sentence as a whole and the meaning of the text) in establishing what simplifications must be used.

[3] See (Power and Williams, 2012) for a discussion of appropriate hedges for values near the extreme points of 0 and 1.

## 6 Conforming with conclusions of prior surveys

The results presented for the system are evaluated in this section for conformance with the conclusions resulting from the empirical studies described in (Bautista et al., 2011b) and (Bautista et al., 2011a).

With respect to the preference for round or common values in simplification (Conclusion 1), the system presented conforms to this preference by virtue of the way in which the list of candidate substitutions is produced by the program. The candidates returned by the program are already restricted to common values of percentages (rounded up) and fractions, so the decision to consider as preferred candidate the one listed first implicitly applies the criteria that leads to this behavior.

With respect to the need to treat differently values in the extreme or central ranges of proportion (Conclusion 2), the system addresses this need by virtue of the actual set of candidates produced by the program in each case. For example, if the original expression is a extreme value like '0.972', the program produces a different candidate substitution ('almost all') that in the central ranges is not considered.

With respect to restricting the use of hedges to situations where loss of precision is incurred (Conclusion 3), the hedge choice rules applied by the system (see Table 1) satisfy this restriction. When *Vr=Vmg* hedges are included in the simplified expression only if they were already present in the original expression.

In addition, the system rounds up any quantities with decimal positions to the nearest whole number whenever the decimal positions are lost during simplification. This functionality is provided implicitly by the program, which presents the rounded up version as the next option immediately following the alternative which includes the decimal positions. For example, if the input proportion is '0.198', some rounded candidate substitutions are calculated as 'almost 20%' or 'less than 20%'.

Finally, the system follows the European guidelines for the production of easy to read information in that it automatically replaces numerical quantities expressed in words with the corresponding quantity expressed in digits.

## 7 Conclusions and future work

The system described in this paper constitutes a first approximation to the task of simplifying numerical expressions in a text to varying degrees of difficulty. The definition of an scale of difficulty of numerical expressions, the identification of rules governing the selection of candidate substitution and the application of hedges constitute important contributions. The empirical evaluation of the system with human experts results in acceptable rates of agreement. The behavior of the system conforms to the conclusions on simplification strategies as applied by humans resulting from previous empirical surveys.

There are different aspects to improve the actual system from the data collected, with a special attention to cases in which the experts disagree. As future work, the syntactic context should be considered to simplify numerical expression, extending the kind of proportion to simplify and treating special cases analyzed in this first version. At the syntactic level, some transformation rules can be implemented from a syntactic analysis. It is important that the meaning of the sentences be preserved regardless of whether part of the sentence is deleted or rewritten by the adaptation rules. In addition, the numerical expression parser and the proportion approximation program could also be studied in order to evaluate the impact of their errors in the final performance.

Our final aim is to develop an automatic simplification system in a broader sense, possibly including more complex operations like syntactic transformations of the structure of the input text, or lexical substitution to reduce the complexity of the vocabulary employed in the text. Additionally we hope to develop versions of the simplification system for other languages, starting with Spanish. Probably the simplification strategies for numbers would be the same but the use of hedge modifiers may be different.

## Acknowledgments

This research is funded by the Spanish Ministry of Education and Science (TIN2009-14659-C03-01 Project), Universidad Complutense de Madrid and Banco Santander Central Hispano (GR58/08 Research Group Grant), and the FPI grant program.

## References


Susana Bautista, Pablo Gervás, and Ignacio Madrid. 2009. Feasibility Analysis for SemiAutomatic Conversion of Text to Improve Readability. In *Proceedings of The Second International Conference on Information and Communication Technologies and Accessibility*, Hammamet, Tunusia, May.

Susana Bautista, Raquel Hervás, Pablo Gervás, Richard Power, and Sandra Williams. 2011a. Experimental identification of the use of hedges in the simplification of numerical expressions. In *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies*, pages 128–136, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Susana Bautista, Raquel Hervás, Pablo Gervás, Richard Power, and Sandra Williams. 2011b. How to Make Numerical Information Accessible: Experimental Identification of Simplification Strategies. In Campos, Pedro and Graham, Nicholas and Jorge, Joaquim and Nunes, Nuno and Palanque, Philippe and Winckler, Marco, editor, *Human-Computer Interaction INTERACT 2011*, volume 6946 of *Lecture Notes in Computer Science*, pages 57–64. Springer Berlin / Heidelberg.

Ann M. Bisantz, Stephanie Schinzing, and Jessica Munch. 2005. Displaying uncertainty: Investigating the effects of display format and specificity. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 47(4):777.

Yvonne Canning. 2000. Cohesive simplification of newspaper text for aphasic readers. In *3rd annual CLUK Doctoral Research Colloquium*.

Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *In Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044.

Clark Christina and Douglas Jonathan. 2010. Young people reading and writing today: Whether, what and why. Technical report, London: National Literacy Trust.

Walter Daelemans, Anja Hothker, and Erik Tjong Kim Sang. 2004. Automatic Sentence Simplification for Subtitling in Dutch and English. In *Proceedings of the 4th Conference on Language Resources and Evaluation*, pages 1045–1048, Lisbon, Portugal.

Siobhan Devlin and John Tait. 1998. *The use of a Psycholinguistic database in the Simplification of Text for Aphasic Readers*. Lecture Notes. Stanford, USA: CSLI.

Nathan Dieckmann, Paul Slovic, and Ellen Peters. 2009. The use of narrative evidence and explicit likelihood by decision makers varying in numeracy. *Risk Analysis*, 29(10).

Geert Freyhoff, Gerhard Hess, Linda Kerr, Elizabeth Menzel, Bror Tronbacke, and Kathy Van Der Veken. 1998. European guidelines for the production of easy-to-read information.

Caroline Gasperin, Lucia Specia, Tiago F. Pereira, and Sandra M. Aluisio. 2009. Learning when to simplify sentences for natural text simplification. In *Proceedings of the Encontro Nacional de Inteligencia Artificial (ENIA)*, pages 809–818, Bento Gonalves, Brazil.

Deeqa Jama and George Dugdale. 2010. Literacy: State of the nation. Technical report, National Literacy Trust.

Arnaldo Candido Jr., Erick Maziero, Caroline Gasperin, Thiago A. S. Pardo, Lucia Specia, and Sandra M. Aluisio. 2009. Supporting the Adaptation of Texts for Poor Literacy Readers: a Text Simplification Editor for Brazilian Portuguese. In *Proceedings of the NAACL/HLT Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42, Boulder, Colorado.

Imtiaz Hussain Khan, Kees Deemter, and Graeme Ritchie. 2008. Generation of refering expressions: managing structural ambiguities. In *Proceedings of the 22nd International Conference on Computational Linguistics(COLING)*, pages 433–440, Manchester.

Himanshu Mishra, Arul Mishra, and Baba Shiv. 2011. In praise of vagueness: malleability of vague information as a performance booster. *Psychological Science*, 22(6):733–8, April.

Ellen Peters, Judith Hibbard, Paul Slovic, and Nathan Dieckmann. 2007. Numeracy skill and the communication, comprehension, and use of risk-benefit information. *Health Affairs*, 26(3):741–748.

Sarah E. Petersen and Mari Ostendorf. 2007. Text Simplification for Language Learners: A Corpus Analysis. In *Proceedings of Workshop on Speech and Language Technology for Education (SLaTE)*.

Richard Power and Sandra Williams. 2012. Generating numerical approximations. *Computational Linguistics*, 38(1).

Qualification and Curriculum Authority. 1999. Mathematics: the National Curriculum for England. Department for Education and Employment, London.

Advaith Siddharthan. 2002. Resolving attachment and clause boundary amgiguities for simplifying relative clause constructs. In *Proceedings of the Student Research Workshop, 40th Meeting of the Association for Computacional Linguistics*.

Advaith Siddharthan. 2003. *Syntactic Simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge.

United Nations. 1994. Standard Rules on the Equalization of Opportunities for Persons with Disabilities. Technical report.

Sandra Williams and Richard Power. 2009. Precision and mathematical form in first and subsequent mentions of numerical facts and their relation to document structure. In *Proc. of the 12th European Workshop on Natural Language Generation*, Athens.

Sandra Williams and Ehud Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proceeding of the 10th European Workshop on Natural Language Generation*, pages 140–147, Aberdeen, Scotland.

Joel Williams, Sam Clemens, Karin Oleinikova, and Karen Tarvin. 2003a. The Skills for Life survey: A national needs and impact survey of literacy, numeracy and ICT skills. Technical Report Research Report 490, Department for Education and Skills.

Sandra Williams, Ehud Reiter, and Liesl Osman. 2003b. Experiments with discourse-level choices and readability. In *In Proceedings of the European Natural Language Generation Workshop (ENLG) and 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL03)*, pages 127–134.

Sandra Williams. 2010. A Parser and Information Extraction System for English Numerical Expressions. Technical report, The Open University, Milton Keynes, MK7 6AA, U.K.