A Hybrid System for Spanish Text Simplification

Stefan Bott
Universitat Pompeu Fabra
C/ Tanger, 122-140
Barcelona, Spain

Horacio Saggion
Universitat Pompeu Fabra
C/ Tanger, 122-140
Barcelona, Spain

David Figueroa
Asi-soft
C/ Albasanz 76
Madrid, Spain

Abstract

This paper addresses the problem of automatic text simplification. Automatic text simplifications aims at reducing the reading difficulty for people with cognitive disability, among other target groups. We describe an automatic text simplification system for Spanish which combines a rule based core module with a statistical support module that controls the application of rules in the wrong contexts. Our system is integrated in a service architecture which includes a web service and mobile applications.

1 Introduction

According to the Easy-to-Read Foundation at least 5% of the world population is functional illiterate due to disability or language deficiencies. Easy access to digital content for the intellectual disabled community or people with difficulty in language comprehension constitutes a fundamental human right (United Nations, 2007); however it is far from being a reality. Nowadays there are several methodologies that are used to make texts easy to read in such ways that they enable their reading by a target group of people. These adapted or simplified texts are currently being created manually following specific guidelines developed by organizations, such as the Asociación Facil Lectura,[1] among others. Conventional text simplification requires a heavy load of human resources, a fact that not only limits the number of simplified digital content accessible today but also makes practically impossible easy access to already available (legacy) material. This barrier is especially important in contexts where information is generated in real time – news – because it would be very expensive to manually simplify this type of "ephemeral" content.

Some people have no problem reading complicated official documents, regulations, scientific literature etc. while others find it difficult to understand short texts in popular newspapers or magazines. Even if the concept of "easy-to-read" is not universal, it is possible in a number of specific contexts to write a text that will suit the abilities of most people with literacy and comprehension problems. This easy-to-read material is generally characterized by the following features:

- The text is usually shorter than a standard text and redundant content and details which do not contribute to the general understanding of the topic are eliminated.[2] It is written in varied but fairly short sentences, with ordinary words, without too many subordinate clauses.
- Previous knowledge is not taken for granted. Backgrounds, difficult words and context are explained but in such a way that it does not disturb the flow of the text.
- Easy-to-read is always easier than standard language. There are differences of level in differ-

[1] http://www.lecturafacil.net

[2] Other providers, for example the Simple English Wikipedia (http://simple.wikipedia.org) explicitly oppose to content reduction. The writing guidelines for the Simple English Wikipedia include the lemma "Simple does not mean short".

NAACL-HLT 2012 Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), pages 75–84,
Montréal, Canada, June 7–8, 2012.

ent texts, all depending on the target group in mind.

Access to information about culture, literature, laws, local and national policies, etc. is of paramount importance in order to take part in society, it is also a fundamental right. The United Nations (2007) "Convention on the Rights of Persons with Disabilities" (Article 21) calls on governments to make all public information services and documentation accessible for different groups of people with disabilities and to encourage the media - television, radio, newspapers and the internet - to make their services easily available to everyone. Only a few systematic efforts have been made to address this issue. Some governments or organisations for people with cognitive disability have translated documents into a language that is "easy to read", however, in most countries little has been done and organizations and people such as editors, writers, teachers and translators seldom have guidelines on how to produce texts and summaries which are easy to read and understand.

1.1 Automatic Text Simplification

Automatic text simplification is the process by which a computer transforms a text for a particular readership into an adapted version which is easier to read than the original. It is a technology which can assist in the effort of making information more accessible and at the same time reduce the cost associated with the mass production of easy texts. Our research is embedded within the broader context of the Simplext project (Saggion et al., 2011).[3] It is concerned with the development of assistive text simplification technology in Spanish and for people with cognitive disabilities. The simplification system is currently under development. Some of the components for text simplification are operational, while other parts are in a development stage. The system is integrated in a larger service hierarchy which makes it available to the users. This paper concentrates on syntactic simplification, as one specific aspect, which is a central, but not the only aspect of automatic text simplification. More concretely, we present a syntactic simplification module, which is based on a hybrid technique: The core of the system is a hand-written computational grammar which reduces syntactic complexity and the application of the rules in this grammar is controlled by a statistical support system, which acts as a filter to prevent the grammar from manipulating wrong target structures. Section 2 describes related work, in the context of which our research has been carried out. Section 3 justifies the hybrid approach we have taken and section 4 describes our syntactic simplification module, including an evaluation of the grammar and the statistical component. Finally, in section 5 we show how our simplification system is integrated in a larger architecture of applications and services.

[3] http://www.simplext.es

2 Related Work

As it has happened with other NLP tasks, the first attempts to tackle the problem of text simplification were rule-based (Chandrasekar et al., 1996; Siddharthan, 2002). In the last decade the focus has been gradually shifting to more data driven approaches (Petersen and Ostendorf, 2007) and hybrid solutions. The PorSimples (Aluísio et al., 2008; Gasperin et al., 2010) project used a methodology where a parallel corpus was created and this corpus was used to train a decision process for simplification based on linguistic features. Siddharthan (2011) compares a rule-based simplification system with a simplification system based on a general purpose generator.

Some approaches have concentrated on specific constructions which are especially hard to understand for readers with disabilities (Carroll et al., 1998; Canning et al., 2000), others focused on text simplification as a help for other linguistic tasks such as the simplification of patent texts (Mille and Wanner, 2008; Bouayad-Agha et al., 2009). Recently the availability of larger parallel or quasi-parallel corpora, most notably the combination of the English and the Simple English Wikipedia, has opened up new possibilities for the use of more purely data-driven approaches. Zhu et al. (2010), for example, use a tree-based simplification model which uses techniques from statistical machine translation (SMT) with this data set.

A recent work, which is interesting because of its purely data-driven setup, is Coster and Kauchak

(2011). They use standard software from the field of statistical machine translation (SMT) and apply these to the problem of text simplification. They complement these with a deletion component which was created for the task. They concentrate on four text simplification operations: *deletion*, *rewording* (lexical simplification), *reordering* and *insertions*. Text simplification is explicitly treated in a similar way to sentence compression. They use standard SMT software, Moses (Koehn et al., 2007) and GIZA++ (Och and Ney, 2000), and define the problem as translating from English (represented by the English Wikipedia) to Simple English (represented by the Simple English Wikipedia). The translation process can then imply any of the four mentioned operations. They compared their approach to various other systems, including a dedicated sentence compression system (Knight and Marcu, 2002) and show that their system outperforms the others when evaluated on automatic metrics which use human created reference text, including BLEU (Papineni et al., 2002). Their problem setting does, however, not include sentence splitting (as we will describe below). Another potential problem is that the metrics they use for evaluation compare to human references, but they do not necessarily reflect human acceptability or grammaticality.

Woodsend and Lapata (2011) use quasi-synchronous grammars as a more sophisticated formalism and integer programming to learn to translate from English to Simple English. This system can handle sentence splitting operations and the authors use both automatic and human evaluation and show an improvement over the results of Zhu et al. (2010) on the same data set, but they have to admit that learning from parallel bi-text is not as efficient as learning from revision histories of the Wiki-pages. Text simplification can also be seen as a type of paraphrasing problem. There are various data-driven approaches to this NLP-task (Madnani and Dorr, 2010), but they usually focus on lexical paraphrases and do not address the problem of sentence splitting, either.

Such data-driven methods are very attractive, especially because they are in principle language independent, but they do depend on a large amount of data, which are not available for the majority of languages.

3 A Hybrid Approach to Text Simplification

There are several considerations which lead us to take a hybrid approach to text simplification. First of all there is a lack of parallel data in the case of Spanish. Within our project we are preparing a corpus of Spanish news texts (from the domain of national news, international news, society and culture), consisting of 200 news text and their manually simplified versions. The manual simplification is time consuming and requires work from specially trained experts, so the resulting corpus is not very big, even if the quality is controlled and the type of data is very specific for our needs. It is also very hard to find large amounts of parallel text from other sources. In order to use data driven techniques we would require amounts of bi-text comparable to those used for statistical machine translation (SMT) and this makes it nearly impossible to approach the problem from this direction, at least for the time being.

But there are also theoretic considerations which make us believe that a rule based approach is a good starting point for automatic text simplification. We consider that there are at least four separate NLP tasks which may be combined in a text simplification setting and which may help to reduce the reading difficulty of a text. They all have a different nature and require different solutions.

- Lexical simplification: technical terms, foreign words or infrequent lexical items make a text more difficult to understand and the task consists in substituting them with counterparts which are easier to understand.

- Reduction of syntactic complexity: long sentences, subordinate structure and especially recursive subordination make a text harder to understand. The task consists in splitting long sentences in a series of shorter ones.

- Content reduction: redundant information make a text harder to read. The task consists in identifying linguistic structures which can be deleted without harming the text grammaticality and informativeness in general. This task is similar to the tasks of automatic summarization and sentence compression.

- Clarification: Explaining difficult concepts reduces the difficulty of text understanding. The task consists in identifying words which need further clarification, selecting an appropriate place for the insertion of a clarification or a definition and finding an appropriate text unit which actually clarifies the concept.

There is at least one task of the mentioned which does not fully correspond to an established machine learning paradigm in NLP, namely the reduction of syntactic complexity. Consider the example (1), an example from our corpus; (2) is the simplification which was produced by our system.

(1) Se trata de un proyecto novedoso y pionero que coordina el trabajo de seis concejalías, destacando las delegaciones municipales de Educación y Seguridad ...

"This is a new and pioneering project that coordinates the work of six councillors, highlighting the municipal delegations Education and Safety ..."

(2) Se trata de un proyecto novedoso y pionero , destacando las delegaciones municipales de Educación y Seguridad ...
Este proyecto coordina el trabajo de seis concejalías.

"This is a new and pioneering project, highlighting the municipal delegations Education and Safety ...
This project coordinates the work of six councillors."

What we can observe here is a split operation which identifies a relative clause, cuts it out of the matrix clause and converts it into a sentence of its own. In the process the relative pronoun is deleted and a subject phrase (*este proyecto* / *this project*) has been added, whose head noun is copied from the matrix clause. It is tempting to think that converting a source sentence A in a series of simplified sentences $\{b_1, \ldots, b_n\}$ is a sort of translation task, and a very trivial one. In part this is true: most words translate to a word which is identical in its form and they happen to appear largely in the same order. The difficult part of the problem is that translation is usually an operation from sentence to sentence, while here the problem setting is explicitly one in which one input unit produces several output units. This also affects word alignment: in order to find the alignment for the word *proyecto* in (1) the alignment learner has to identify the word *proyecto* in two sentences in (2). The linear distance between the two instances of this noun is considerable and the sentences in which two alignment targets occur are not even necessarily adjacent. In addition, there may be multiple occurrences of the same word in the simplified text which are not correct targets; the most apparent case are functional words, but even words which are generally infrequent may be used repeatedly in a small stretch of text if the topic requires it (in this paragraph, for example, the word *translation* occurs 4 times and the word *sentence* 5 times). While a machine can probably learn the one-to-may translations which are needed here, a non-trivial extension of the machine-translation setting is needed and the learning problem needs to be carefully reformulated. Applying standard SMT machinery does not seem to truly address the problem of syntactic simplification. In fact, some approaches to SMT try use text simplification as a pre-process for translation; for example Poornima et al. (2011) apply a sentence splitting module in order to improve translation quality.

On the other hand, other sub-task mentioned above can be treated with data driven methods. Lexical simplification requires the measurement of lexical similarity, combined with word sense disambiguation. Content reduction is very similar to extractive summarization or sentence compression and the insertion of clarifications can be broken down into three learnable steps: identification of difficult words, finding an insertion site and choosing a suitable definition for the target word.

4 Syntactic Simplification

We are developing a text simplification system which will integrate different simplification modules, such as syntactic simplification, lexical simplification (Drndarevic and Saggion, 2012) and content reduction. At the moment the most advanced module of this system is the one for syntactic simplification. In (Bott et al., 2012) we describe the functioning of the simplification grammar in more detail.

For the representation of syntactic structures we

use dependency trees. The trees are produced by the Mate-tools parser (Bohnet, 2009) and the syntactic simplification rules are developed within the MATE framework (Bohnet et al., 2000). MATE is a graph transducer which uses hand written grammars. For grammar development we used a development corpus of 282 sentences.

The grammar mainly focuses on syntactic simplification and, in particular, sentence splitting. The types of sentence splitting operations we treat at the moment are the following ones:

- Relative clauses: we distinguish between simple relative clauses which are only introduced by a bare relative pronoun (e.g. *a question which is hard to answer*) and complex relative clauses which are introduced by a preposition and a relative pronoun (e.g. *a question to which there is no answer*)
- Gerundive constructions and participle constructions (e.g. *the elections scheduled for next November*)
- Coordinations of clauses (e.g.*[the problem is difficult] and [there is probably no right answer]*) and verb phrases (e.g. *The problem [is difficult] and [has no easy solution]*).
- Coordinations of objects clauses (e.g. . . . *to get close to [the fauna], [the plant life] and [the culture of this immense American jungle region]*)

We carried out a evaluation of this grammar, which is resumed in Table 1. This evaluation looked at the correctness of the output. Many of the errors were due to wrong parse trees and and the grammar produced an incorrect output because the parsed input was already faulty. In the case of relative clauses nearly 10% occurred because of this and in the case of gerundive construction 37% of the errors belonged into that category. We also found that many of the syntactic trees are ambiguous and cannot be disambiguated only on the basis of morphosyntactic information. A particular case of such ambiguity is the distinction between restrictive and non-restrictive relative clauses. Only non-restrictive clauses can be turned into separate sentences and the distinction between the two types is usually not marked by syntax in Spanish[4]. Error analysis showed us that 57.58% of all the errors related to relative clauses were due to this distinction. A further 18.18% of the error occurred because the grammar wrongly identified complement clauses as relative clauses (in part because of previous parsing errors).

For this reason, and according to our general philosophy to apply data-driven approaches whenever possible, we decided to apply a statistical filter in order to filter out cases where the applications of the simplification rules lead to incorrect results. Figure 1 shows the general architecture of the automatic simplification system, including the statistical filter. The nucleus of the system in its current state is the syntactic simplification system, implemented as a MATE grammar, which consists of various layers.

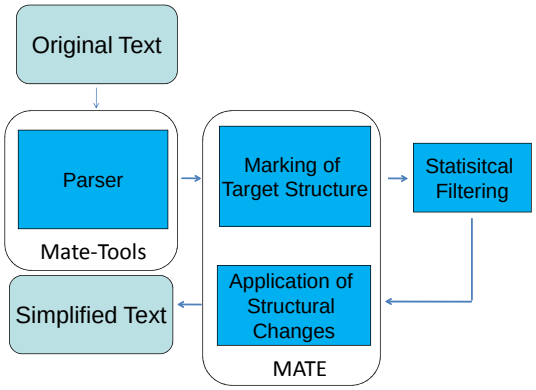

Figure 1: The architecture of the simplification system

Syntactic simplification is carried out in three steps: first a grammar looks for suitable target structures which could be simplified. Such structures are then marked with an attribute that informs subsequent levels of the grammar. After that the statistical filter applies and classifies the marked target structures according to whether they should be changed or not. In a third step the syntactic manipulations themselves are carried out. This can combine deletions, insertions and copying of syntactic nodes or subtrees.

[4]In English it is mandatory to place non-restrictive relative clauses between commas, even if many writers do not respect this rule, but in Spanish comma-placement is only a stylistic recommendation.

Operation	Precision	Recall	Frequency
Relative Clauses (all types)	39.34%	0.80%	20.65%
Gerundive Constructions	63.64%	20.59%	2.48%
Object coordination	42.03%	58.33%	7.79%
VP and clause coordination	64.81%	50%	6.09%

Table 1: Percentage of right rule application and frequency of application (percentage of sentences affected) per rule type

4.1 Statistical Filtering

Since the training of such filters requires a certain amount of hand-annotated data, so far we only implemented filters for simple and complex relative clauses. These filters are implemented as binary classifiers. For each structure which the grammar could manipulate, the classifier decides if the simplification operation should be carried out or not. In this way, restrictive relative clauses, complement clauses and other non-relative clause constructions should be retained by the filter and only non-restrictive relative clauses are allowed to pass.

For the training of the filters we hand annotated a selection of sentences which contained the relevant type of relative clauses (150 cases for simple and 116 for complex). The training examples were taken from news texts published in the on-line edition of an established Spanish newspaper. The style in which these news were written was notably different from the news texts of the corpus we are developing in within our project, in that they were much more complex and contained more cases of recursive subordination. The annotators reported that some of the sentences had to be re-read in order to fully understand them; this is not uncommon in this type of news which may contain opinion columns and in-depth comments.

In our classification framework we consider one set of contextual features arising from tokens surrounding the target structure to be classified[5] – the relative pronoun marked by the simplification identification rules. This set is composed of, among others, the position of the target structure in the sentence; the parts of speech tags of neighbour token; the depth of the target in a dependency tree; the dependency information to neighbour tokens, etc.

Linguistic intuitions such as specific constructions which, according to the Spanish grammar, could be considered as indicating that the simplification can or cannot take place. These features are for example: the presence of a definite or indefinite article; the presence of a comma in the vicinity of the pronoun; specific constructions such as *ya que* (since), *como que* (as), etc. where *que* is not relative pronoun; context where *que* is used as a comparative such as in *más....que* (more... than); contexts where *que* is introducing a subordinate complement as in *quiero que* (I want that ...); etc. While some of these features should be implemented relying on syntactic analysis we have relied for the experiments reported here on finite state approximations implementing all features in regular grammars using the GATE JAPE language (Cunningham et al., 2000; Maynard et al., 2002). For other learning tasks such as deciding for the splitting of coordinations or the separation of participle clauses we design and implement specific features based on intuitions; contextual features remain the same for all problems.

The classification framework is implemented in the GATE system, using the machine learning libraries it provides (Li et al., 2005). In particular, we have used the Support Vector Machines learning libraries (Li and Shawe-Taylor, 2003) which have given acceptable classification results in other NLP tasks. The framework allows us to run cross-validation experiments as well as training and testing.

Table 2 shows the performance of the statistical filter in isolation, i.e. the capacity of the filter alone to distinguish between good and bad target structures for simplification operations. The in-domain performance was obtained by a ten-fold cross classification of the training data. The out-of-domain evaluation was carried out over news texts from our own corpus, the same collection we used for the

[5] A 5 words window to the left and to the right.

Figure 2: A simplified news text produced by the service on a tablet computer running Android

Este miércoles las personas con Sindrome de Down celebran si día mundial . En España , hay más de 34 .000 personas con esta discapacidad . esta discapacidad ocurre en uno de cada 800 nacimientos .

El Sídrome de Down es un trastorno genético . este trastorno causa la presencia de una copia extra del cromosoma 21 en vez de los dos habituales (trisomía del par 21) . La consecuencia es un grado variable de discapacidad cognitiva y unos rásgos físicos particulares y reconocibles .

Se trata de la causa más frecuente de discapacidad cognitiva psíquica congénita y debe su nombre a John Langdon Haydon Down . este Landgdon fue el primero en describir esta alteración genética en 1866 . Siegue sin conocerse con exactitud las causas . estas causas provocan el exceso cromosómico , aúnque se relaciona estadística mente con madres de más de 35 años .

Table 4: The simplified text shown in figure2

evaluation of the grammar and the combination of the grammar with the statistical filter. The performance is given here as the overall classification result. Table 3 shows the performance of the grammar with and without application of the filter.[6]

4.2 Discussion

We can observe that the statistical filters have a quite different performance when they are applied in-domain and out-of-domain (cf. Table 2), especially in the case of simple relative clauses. We attribute this to the fact that the style of the texts which we used for training is much more complicated than the texts which we find in our own corpus. The annotators commented that many relative clauses could not turned into separate sentences because of the overall complexity of the sentence. This problem seems to propagate into the performance of the combination of the grammar with the filter (cf. Table 3). The precision improves with filtering, but the recall drops even more. Again, we suspect that the filter is very restrictive because in the training data many relative clauses were not separable, due to the overall sentence complexity which is much lesser in the corpus from which the test data was taken. For the near future we plan to repeat the experiment with annotated data which is more similar to the test set. The performance in the case of complex relative clauses is much better. We attribute the difference between simple and complex relative clauses to the fact that the complex constructions cannot be confounded with other, non-relative, constructions, while in the case of the simple type this danger is considerable. The somewhat unrealistic value of 100% is a consequence of the fact that in the part of the corpus we annotated complex relative clauses were not very frequent. We took some additional cases from our corpus into consideration, evaluating more cases from the corpus where the corresponding rule was applied[7] and the value dropped to slightly over 90%.

5 Integration of the Simplification System in Applications

As we have mentioned in the introduction, our text simplification system is integrated in a larger service and application setting. Even if some modules of the system must still be integrated, we have an operative prototype which includes a mobile application and a web service.

In the context of the Simplext project two mobile applications have been developed. The first one runs on iOS (developed by Apple Inc. for its devices: Iphone, Ipad and Ipod touch), and the other one on Android (developed by Google, included in many different devices). These applications allow

[6]The results here are not fully comparable to Table 1, because in order to evaluate the filter, we did not consider parse errors, as we did in the previous evaluation.

[7]For these cases we could not calculate recall because this would have implied a more extensive annotation of all the sentences of the part of the corpus from which they were taken.

Operation	Precision	Recall	F-score
Simple Relative Clauses (in domain)	85.41%	86.77%	86.06%
Complex Relative Clauses (in domain)	70.88%	71.33%	71.10 %
Simple Relative Clauses (out of domain)	76.35%	76.35%	76.35%
Complex Relative Clauses (out of domain)	90.48%	85.71%	88.10%

Table 2: The performance of the statistical filter in isolation

Operation	Precision	Recall	F-score
Simple Relative Clauses (Grammar)	47.61%	95.24%	71.43%
Complex Relative Clauses (Grammar)	62.50%	55.56%	59.02%
Simple Relative Clauses (Grammar + Filter)	59.57%	66.67%	63.12%
Complex Relative Clauses (Grammar + Filter)	100%	55.56%	77.78%

Table 3: The performance of grammar and the statistical filter together

to read news feeds (RSS / Atom) from different sources through a proxy that provide the language simplification mechanism. The mobile applications are basically RSS/Atom feed readers, with simplification capabilities (provided by the service layer). Both applications work the same way and allow to the user functionalities as keeping a list of favourite feeds, adding and removing feeds, marking content as favourite and showing the simplified and original versions of the content. Also a web service was created, which works in a similar way for RSS and Atom feeds and allows to simplify the text portion of other publicly available websites.

Figure 2 shows a screen capture of the mobile application running in a Android tablet, displaying a simplification example of a text taken from a news website. The display text of this image is reproduced in Table 4 for better readability. The text itself is too long for us to provide a translation, but it can be seen that many sentences have been split. Also a series of minor problems can be seen, which we will resolve in the near future: The first word of a sentence is still in lower case and the head noun of the named entity *John Langdon Haydon Down* was not correctly identified.

6 Conclusions

Automatic text simplification is an Assistive Technology which help people with cognitive disabilities to gain access to textual information. In this paper we have presented a syntactic simplification module of a automatic text simplification system which is under development. We have presented arguments for the decision of using a hybrid strategy which combines a rule-based grammar with a statistical support component, we have described the implementation of this idea and have given a contrastive evaluation of the grammar with and without statistical support. The simplification system we described here is integrated in a user-oriented service architecture with mobile applications and web services. In future work we will further enhance the system and integrate new components dedicated to other simplification aspects, such as lexical simplification and content reduction.

Acknowledgements

The research described in this paper arises from a Spanish research project called Simplext: An automatic system for text simplification (`http://www.simplext.es`). Simplext is led by Technosite and partially funded by the Ministry of Industry, Tourism and Trade of the Government of Spain, by means of the National Plan of Scientific Research, Development and Technological Innovation (I+D+i), within strategic Action of Telecommunications and Information Society (Avanza Competitiveness, with file number TSI-020302-2010-84). We are grateful to fellowship RYC-2009-04291 from Programa Ramón y Cajal 2009, Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, Spain.

References

Sandra M. Aluísio, Lucia Specia, Thiago Alexandre Salgueiro Pardo, Erick Galani Maziero, and Renata Pontin de Mattos Fortes. 2008. Towards brazilian portuguese automatic text simplification systems. In *ACM Symposium on Document Engineering*, pages 240–248.

Bernd Bohnet, Andreas Langjahr, and Leo Wanner. 2000. A development environment for MTT-based sentence generators. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*.

Bernd Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 67–72, Boulder, Colorado. Association for Computational Linguistics.

Stefan Bott, Horacio Saggion, and Simon Mille. 2012. Text simplification tools for spanish. In *Proceedings of the LREC-2012*, Estambul, Turkey.

Nadjet Bouayad-Agha, Gerard Casamayor, Gabriela Ferraro, and Leo Wanner. 2009. Simplification of patent claim sentences for their paraphrasing and summarization. In *FLAIRS Conference*.

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 2000. Cohesive generation of syntactically simplified newspaper text. In *TSD*, pages 145–150.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *In Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *COLING*, pages 1041–1044.

William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of Text-To-Text Generation*, Portland, Oregon. Association for Computational Linguistics.

H. Cunningham, D. Maynard, and V. Tablan. 2000. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November.

Biljana Drndarevic and Horacio Saggion. 2012. Towards automatic lexical simplification in spanish: an empirical study. In *NAACL 2012 Workshop on Predicting and Improving Text Readability for Target Reader Populations*, Montreal, Canada.

Caroline Gasperin, Erick Galani Maziero, and Sandra M. Aluísio. 2010. Challenging choices for text simplification. In *PROPOR*, pages 40–50.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107, July.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Y. Li and J. Shawe-Taylor. 2003. The SVM with Uneven Margins and Chinese Document Categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, Singapore, Oct.

Yaoyong Li, Katalina Bontcheva, and Hamish Cunningham. 2005. Using Uneven Margins SVM and Perceptron for Information Extraction. In *Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*.

N. Madnani and B.J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Diana Maynard, Valentin Tablan, Hamish Cunningham, Cristian Ursu, Horacio Saggion, Katalina Bontcheva, and Yorik Wilks. 2002. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274.

Simon Mille and Leo Wanner. 2008. Making text resources accessible to the reader: The case of patent claims. Marrakech (Marocco), 05/2008.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. pages 440–447, Hongkong, China, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sarah E. Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *In Proc. of Workshop on Speech and Language Technology for Education*.

C. Poornima, V. Dhanalakshmi, K.M. Anand, and KP Soman. 2011. Rule based sentence simplification for english to tamil machine translation system. *International Journal of Computer Applications*, 25(8):38–42.

H. Saggion, E. Gómez Martínez, E. Etayo, A. Anula, and L. Bourg. 2011. Text simplification in simplext. making text more accessible. *Procesamiento de Lenguaje Natural*, 47(0):341–342.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *In LREC'02: Proceedings of the Language Engineering Conference*, pages 64–71.

Advaith Siddharthan. 2011. Text simplification using typed dependencies: A comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, pages 2–11, September.

United Nations. 2007. Convention on the rights of persons with disabilities. `http://www2.ohchr.org/english/law/disabilities-convention.htm`.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–420.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of The 23rd International Conference on Computational Linguistics*, pages 1353–1361, Beijing, China, Aug.