



Proceedings of the 5th International Conference on Software Development and  
Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2013)

## Integration of lexical and syntactic simplification capabilities in a text editor

Raquel Hervás<sup>a,\*</sup>, Susana Bautista<sup>a</sup>, Marta Rodríguez<sup>a</sup>, Teresa de Salas<sup>a</sup>, Ana Vargas<sup>a</sup>,  
Pablo Gervás<sup>a</sup>

<sup>a</sup>Facultad de Informática, Universidad Complutense de Madrid, c/ Pfor. José G<sup>a</sup> Santesteban, 28040 SPAIN

---

### Abstract

Nowadays, with the rapid development of new technologies, it is very difficult to be an active member of the society unless one is proficient in reading. Unfortunately, there are a lot of people who experience reading or writing problems. Experts who create materials for these groups of people have few user-friendly tools at their disposal in order to prepare easy-to-read texts. The present work has been done with the aim of providing support for text simplification experts who prepare easy-to-read texts in English, which is why different Natural Language Processing tools have been integrated in an already existing text editor.

© 2013 The Authors. Published by Elsevier B.V.

Selection and/or peer-review under responsibility of Scientific Programme Committee of the 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2013)

*Keywords:* Text simplification; Easy-to-read; Lexical simplification; Syntactic simplification; Natural Language Processing.

---

### 1. Introduction

The ability to understand a piece of writing significantly boosts readers' confidence, in the way that it allows them to gain new insights and exercise control over their own lives. However, numerous people lack a sufficiently developed ability to read fluently, write, or understand a written text. As a result, the way a text is written or information is presented may prevent these people from taking part in socioeconomic activities of their society. It is for this reason that various associations work with organisations, publishing houses, and public institutions, providing instruction on how to write texts that follow easy-to-read criteria. Nevertheless,

---

\* Corresponding author. Tel.: +34 91 394 75 99; fax: +34 91 394 75 47.

*E-mail address:* [raquelhb@fdi.ucm.es](mailto:raquelhb@fdi.ucm.es).

there have been few systematic attempts to research this topic, and as a result, both organisations and individuals such as editors, writers, teachers, or translators experience difficulties when adapting a piece of writing to easy-to-read standards. Whether a text is easy to read and understand depends significantly on the ability and experience of the reader. Therefore, the concept of easy-to-read texts cannot be universal and it is impossible to prepare a piece of writing that meets the needs of all the individuals with reading and comprehension problems. That is why it is important to help experts who adapt reading material by allowing them to approach the task in a personalised manner, paying attention to specific needs of a particular group of readers.

Consequently, the present work is mainly intended for people who create easy-to-read texts by simplifying existing pieces of writing which are more difficult to understand. There is a variety of Natural Language Processing (NLP) tools that could help these experts, such as dictionaries from which definitions or synonyms of complicated words can be extracted; applications that apply syntactic constituent analysis or provide a syntactic dependency tree for a given sentence, which in turn allows the user to explore which parts of a sentence may be simplified; etc. One problem that text simplification experts tend to encounter when using these tools is the fact that they lack a user-friendly interface, and they often require some background knowledge of computer science and programming. In an attempt to offer a solution to this problem, we have created an application that encompasses some of previously existing tools so that they can be easily used by experts who work on creating easy-to-read texts. With that aim, we have developed a flexible structure based on an existing text editor. It includes all the basic components that should be made available to a text simplification expert, and additionally, it enables easy integration of specific tools used for different text simplification tasks (for example, substituting a word with a synonyms, splitting complex sentences into simpler ones, etc.).

## **2. Related work**

In this section, we introduce some notions relevant for the field in which our work is situated, the existing solutions in the field, as well as the tools we have used in the process of implementing this work.

### *2.1. Easy to read*

Easy to Read is a movement that aims to create clearly written materials which foster efficient communication and can be read and understood by people with reading comprehension difficulties. According to the European Guidelines for the Production of Easy-to-Read Information [1], an easy-to-read text is one that contains only the most relevant information, expressed and presented in the most direct manner, so that its content may be understood by as many people as possible.

Easy-to-read texts are intended for individuals whose reading ability is lower than that of an adult native speaker. These individuals account for 30% of the population in the developed world. We can divide them into two major groups [2]: people with temporary reading difficulties (immigrants who have not yet achieved proficiency in the language of the receiving country, people with limited cultural information, primary school children who are learning to read, etc.); and people with permanent reading difficulties (due to physical disabilities, such as hearing or visual impairment, or psychological disabilities such as dyslexia, aphasia, etc.)

The European Easy-to-Read Guidelines [1] help create easy-to-read texts and provide direction for the reader in terms of content, language, and text organisation. Despite their general character, and taking into account the fact that Easy to Read is not a universal concept, these guidelines are more easily adapted to the needs of people with disabilities than to those with temporary reading difficulties. Four general recommendations are offered: using simple and direct language; expressing a single idea per sentence; avoiding technical terms, abbreviations, and initials; and organising the text in a clear and coherent manner. It is

important to pay attention to how a text is structured, since it has to be organised in a clear and coherent way. Unnecessary ideas, words, clauses, or sentences should be avoided or eliminated. Images, charts, and symbols should be an accompanying aid. The aim is to create a piece of writing that can be easily read and understood by as many people as possible.

## 2.2. Text simplification projects

The Portuguese project PorSimples [3] offers a technological solution that helps people with cognitive disabilities access information. The project offers two systems for two groups of target users. On the one hand, an editing tool is proposed, which helps writers create and approve simplified texts. On the other hand, the authors suggest a system that helps people with disabilities access certain web content, such as government texts or general news; as a result, digital inclusion and accessibility are promoted.

Within the lexical simplification paradigm the PSET project [4] stands out. It is aimed at simplification of news texts in English for people with aphasia, which is achieved by substituting difficult words with their simpler synonyms extracted from WordNet [5]. To determine word difficulty, the Oxford Linguistic Database is consulted [6]. Caseli et Al. [7] created a parallel corpus in Portuguese and extracted lexical simplification operations applied by human annotators. They used a list of simple words and a list of discourse markers as tools to assist them in conducting synonym substitution.

The main objective of the Spanish Simplext project [8] is to build an automatic text simplification system. The project promotes technological inclusion of people with limited cognitive abilities by means of automatic transformation of content based on easy-to-read principles.

There are numerous other studies conducted on other languages [9, 10, 11, 12, 13, 14] and approaches based on analysis of parallel corpora consisting of an original text and its manual simplification [15, 16].

## 2.3. Natural Language Processing tools

There are many Natural Language Processing (NLP) tools that may be useful for the human editor adapting a text to easy-to-read criteria through simplification. However, these tools are not necessarily easy to use as they require a certain level of expertise which editing experts usually lack. What follows is a description of the most important tools used in this work.

**OpenNLP.** OpenNLP<sup>a</sup> is a set of tools based on machine learning techniques for Natural Language Processing. It provides aid for the most common NLP tasks such as sentence detection, tokenisation, part-of-speech (POS) tagging, and others.

**WordNet.** WordNet [5] is a vast lexical database in English. Nouns, verbs, adjectives, and adverbs are grouped into sets of synonyms denominated synsets. Every synset conveys a different meaning of a given word. A useful interface for consulting WordNet which can be integrated in external code is LightWeight WordNet [17].

**MorphAdorner.** MorphAdorner<sup>b</sup> is a set of Java applications created by Northwestern University and divided into smaller, independent applications that can be used separately or in conjunction with one another.

---

<sup>a</sup> <http://opennlp.apache.org/> [05-09-13]

<sup>b</sup> <http://morphadorner.northwestern.edu/morphadorner/> [05-09-13]

Although the range of NLP tools offered by MorphAdorner is quite wide, in this work we have only used the pluraliser, in order to work with plural and singular forms of the words we wish to simplify.

### 3. Extending a text editor with text simplification capabilities

The work presented here aims at providing support for text simplification in English. We have decided to implement it in a flexible manner using jEdit<sup>c</sup>, an existing text editor, and more specifically by using a series of plugins. We therefore suggest some general features used for all the plugins related to the simplification process, so that any new simplification operation can be implemented as a separate plugin but using a common set of auxiliary functionalities. Such a model allows the simplification expert to enable or to turn off the transformations based on their immediate needs, while the basic components relevant for all simplification operations are always running.

jEdit is an open source plain text editor with multiplatform support, and it can be easily configured and personalised. In order to extend its functionality, jEdit enables the introduction of different plugins, which can be downloaded from its official web site or directly from the software creator page. In addition to that, there is a project in Source-Forge called jEdit Plugin Central<sup>d</sup>, which allows any user to create and publish their own plugins.

#### 3.1. Auxiliary components for the simplification process

The basic components intended to be used by all plugins include an auxiliary module that shows the information in a user-friendly manner, and the creation of a log where all the changes applied by the editor are stored.

**BasicHighlighter.** In order to mark certain parts of the text for the user or to allow them to point out possible simplifications, it is necessary to highlight parts of the text in jEdit. With the aim of implementing this component, we have developed an auxiliary module as a simpler version of an already existing plugin (Highlight<sup>e</sup>), which enables highlighting words or regular expressions in different colours.

**Log.** Our system also stores a log in an XML format with all the simplification operations applied through different plugins. The log has been implemented with its possible future use in mind, in the way that it stores all the necessary information to study the applied simplification steps at some point in the future. The order in which changes have been applied is also important, as it allows us to analyse the way in which an expert simplifies a text, and enables future automation of this process. Including this kind of information in the system opens up many possibilities in the future and it can be used as a starting point for developing an automatic text simplification application.

#### 3.2. Deciding the desired functionalities through the analysis of simplified texts

In order to determine which simplification operations may be useful to the expert preparing an easy-to-read text, we have conducted a preliminary study with native English speakers who were asked to simplify short texts as they found best. This study has provided us with various preliminary conclusions, although we intend

---

<sup>c</sup> <http://www.jedit.org/> [05-09-13]

<sup>d</sup> <http://plugins.jedit.org/> [05-09-13]

<sup>e</sup> <http://plugins.jedit.org/plugins/?Highlight> [05-09-13]

to conduct in-depth analysis of applied simplification operations in the future. Table 1 summarises the simplification operations we have identified.

**Table 1.** Types of simplification operations found in the texts.

<b>Simplification</b>	<b>Total (Frequency)</b>
Rewriting	28 (26.17%)
Synonym (verb)	25 (23.36%)
Synonym (noun)	20 (18.69%)
Syntactic	19 (17.76%)
Deletion of text	8 (7.48%)
Synonym (other)	7 (6.54%)
<b>Total</b>	<b>107</b>

The most commonly applied operation is the rewriting or restructuring of the text, trying to express it in a simpler way. For example, the sentence: “*According to the Italian navigation code, a captain who abandons a ship in danger can face up to twelve years in prison*” was substituted by “*The rules of the Italian boat workers says that a captain who leaves a ship in a problem can go to jail for as many as twelve years*”, that express the same ideas with different words and syntactic structures.

At the lexical level a very frequent operation is the substitution of nouns and verbs considered complicated with equivalent expressions or clauses, in particular synonyms. For example, the clause “*who was spotted by Coast Guard officials*” was substituted with “*who was seen by boat workers*”; and “*the chaotic and terrifying evacuation was underway*” was simplified as “*during the crazy and scary rescue*”. In the sentence “*The captain abandoned the stricken liner before all the passengers had escaped*” the participants in the study chose to substitute various words with simpler synonyms, the resulting sentence being “*The captain left the damaged ship before all the people on the ship had left*”. In other texts we observed the same type of simplification. For example, the verb form “*turned up*” was substituted with “*appeared*”, and “*got in touch*” was changed into “*called him*”.

From a syntactical point of view the analysis is more complex. Syntactical simplifications performed by participants do not present very specific patterns. However, there are many sentences that were simplified by dividing the text into sentences representing only one idea. For example, the sentence “*An Indonesian girl swept away in 2004 tsunami has been reunited with her parents seven years on, the family say*” was transformed into “*In 2004 a tsunami swept away an Indonesian girl. She was separated from her family. After seven years her family says that she was reunited with them*”. This kind of simplification was also found in the sentence “*The little mermaid who was the youngest and had a wonderful voice spent her time thinking on the surface, the world of the men*”, which was simplified like “*The little mermaid was the youngest and had a wonderful voice. She spent her time thinking about the world of the men*”.

Another common simplification is the deletion of irrelevant or less important data, specially adjectives and circumstantial complements that do not add essential information. For example, in a sentence “*the ship’s Italian owner*” was substituted with “*the owner of the ship*” omitting the adjective “*Italian*”.

In this work we concentrate on lexical and syntactical simplifications. Other simplifications like the rewriting or deletion of text are more difficult to be partially automated and mostly depend on the editor’s expertise. Since one of the most common simplification operation is substitution of nouns and verbs, for the current version of our application we have developed a lexical plugin that suggests synonyms for selected nouns and verbs. From a syntactical point of view we have developed a syntactical plugin that splits complex coordinate sentences into pieces so each resulting sentence expresses fewer ideas.

### 3.3. A lexical simplification plugin: Synonyms

The lexical plugin *Synonyms* substitutes nouns or verbs in infinitive that the user finds difficult with their simpler synonyms. With that aim, a number of candidate synonyms are suggested, and if the user is not satisfied with the choice, they are given the opportunity to edit the target word. In order to be able to use the plugin, the document to be simplified needs to be opened in jEdit.

We choose a word from the text and we look for its synonyms by clicking *Plugins > Synonyms > Get Synonyms*. OpenNLP is then used to determine the grammatical category (POS) of the selected word. As a result, we can tell whether the word is a noun or a verb, information needed when looking it up in the dictionary. We also use the sentence detector in OpenNLP to detect the original sentence and the sentence obtained after the selected word has been substituted with its synonym, and this information is stored in the log.

Our plugin uses WordNet to determine the meaning and the synonyms of the word to be simplified. WordNet lists different meanings of a given word, and based on that, different sets of synonyms (synsets). For the time being, we access the first synset in order to use its content as candidate synonyms (in WordNet the first synset is always the most frequent meaning of the word). However, disambiguation operations will be added in the future in order for the application to be able to choose the right synset in every instance.

One problem with using WordNet is that it does not contain nouns in plural. The user may select a plural word form, which returns no results when looked up in WordNet. In order to tackle this problem, a pluraliser function has been added using MorphAdorner. When OpenNLP determines that the selected word is in plural, the pluraliser returns its singular form so that its meaning and synonyms can be looked up in WordNet. Once the synonym is obtained, the pluraliser returns its plural form again. This step further simplifies the process of text simplification, since manually adjusting agreement in the text is avoided.

If the original word has synonyms, the user is shown a panel such as the one in Figure 1. It can be observed from the example in the figure that the word we want to simplify is “*automobile*” and that all instances of this word in the text are highlighted. The definition of the chosen word is shown in the panel.

For every instance of the word, starting with the selected word, we can choose among the following actions: (1) leave the word unchanged by clicking *Next*, (2) replace the word with one of the synonyms from the list, by selecting the chosen synonym and clicking *Replace*, and (3) edit the word ourselves by choosing the option *Edit in your own words*, and introducing our synonym in the space provided. As in the previous option, we apply substitution by clicking *Replace*. The option *Replace All* replaces all the highlighted instances of the word with the chosen option. One can navigate between different word instances by clicking *Next* or *Back*, or by going forward or backwards until the beginning or the end of the text are reached, where a new search can be started.

Apart from that, the selected instances that have not been substituted can be left highlighted in order to go back to them later. Similarly, if the user decides to leave the text as it is, they can finish the simplification of the word by clicking *Finish processing*.

The *Synonyms* plugin log saves all the changes made to the original document. In the context of lexical simplification, it is interesting to save the substituted word, the substituting word, and the surrounding context. Suggested synonyms are also saved, as well as any corrections made by the user and the order in which different substitutions took place within the text.

By definition an **automobile** or car is a wheeled vehicle that carries passengers. The **automobile** as we know it was not invented in a single day by the **automobile** reflects an evolution that took place worldwide.

It is estimated that over 100,000 patents created the modern **automobile** that occurred along the way to producing the modern car; and with articles, biographies, timelines, and photo galleries related to inventors.

The **automobile** as we know it was not invented in a single day by the **automobile** reflects an evolution that took place worldwide. It created the modern **automobile**. However, we can point to the many Starting with the first theoretical plans for a motor vehicle the Leonardo da Vinci and Isaac Newton.

In 1769, the very first self-propelled road vehicle was a military and mechanic, Nicolas Joseph Cugnot (1725 - 1804). Cugnot used a built under his instructions at the Paris Arsenal by mechanic Breton to haul artillery at a whopping speed of 2 1/2 mph on only three ten to fifteen minutes to build up steam power. The steam engine of the vehicle and placed in the front (see engraving above). The a steam-powered tricycle that carried four passengers.

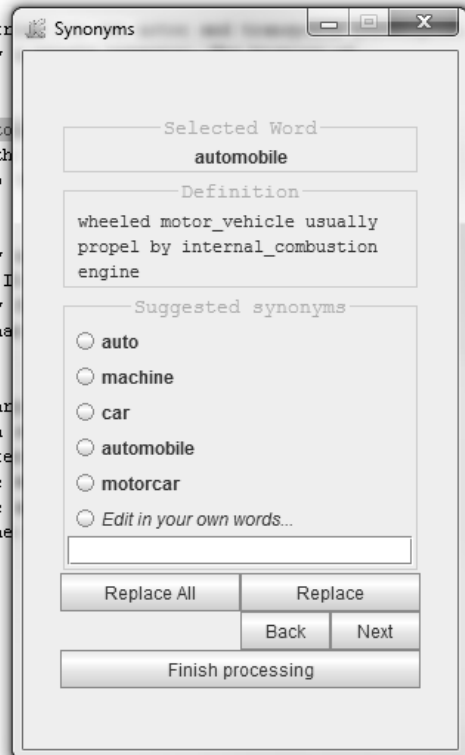


Fig. 1. A screenshot of the lexical plugin panel

### 3.4. A syntactic simplification plugin: *Split*

The syntactic plugin *Split* provides support in the syntactic simplification process, splitting complex coordinate sentences into simpler ones. In order to be able to use the plugin, the document to be simplified needs to be opened in jEdit. If the plugin is loaded in the editor, it has to be activated (*Plugins > Split > Enable to Split*) in order to be used. After activation, all occurrences of the ‘and’ conjunction are highlighted so that they can be easily located in the text.

When the user selects a conjunction, the system splits the compound sentence into different clauses. The conjunctions are detected using the tokenisation and part-of-speech annotation functionalities of OpenNLP. In addition, the sentence in which the selected conjunction is located is parsed in order to obtain the syntactic tree. Two different structures are possible in the sentences that can be processed by the plugin:

- The clauses that form the coordinate sentence do not share syntactic elements. This is the case with the sentences where the two clauses have been joined by the conjunction ‘and’, as in “*Peter is running on the beach and a girl is eating an apple*”. Starting with the syntactic tree from the parser, we create separate trees, each corresponding to one of the coordinate clauses. Thus after processing,



the sentence from the previous example becomes: “*Peter is running in the beach. A girl is eating an apple*”.

- The clauses that form the coordinate sentence share the subject, as in “*Peter was running and arrived late*”. Starting with the syntactic tree from the parser we create separate trees, but this time the subject of the first sentence is duplicated in each of the new sentences. Therefore, the resulting text for the previous example is “*Peter was running. Peter arrived late*”.

If the compound sentence is tractable by the system, the user is shown a panel as the one in Figure 2. The original sentence and the suggestion made by the system are presented. The final sentence can be edited by the user. The text is changed once the button *Apply* is pressed. If the user decides to leave the text as it is, the option *Plugins > Split > Disable to Split* deactivates the plugin and the conjunctions are no longer highlighted. The *Split* plugin log saves all the changes made to the original document. In the context of syntactic simplification, it is interesting to save the original and new sentences, the sentence suggested by the system, and if the suggestion was edited by the user or not.

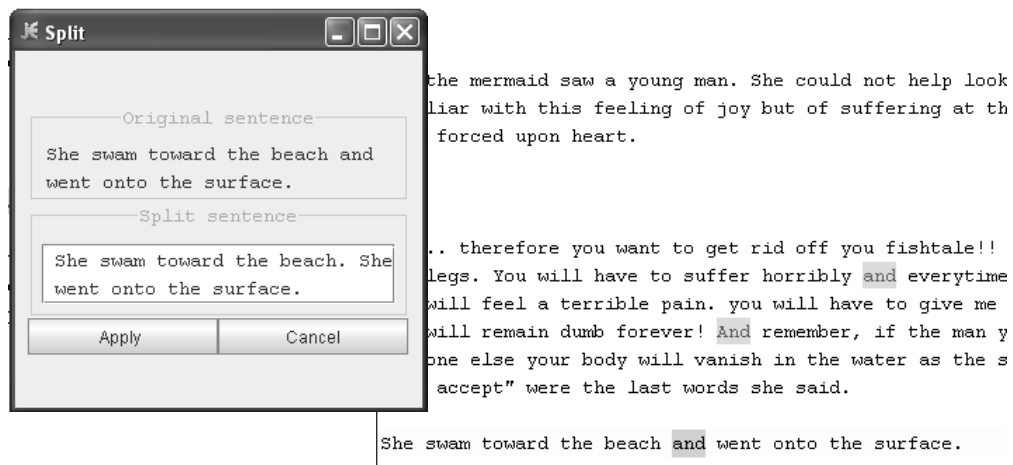


Fig. 2. A screenshot of the syntactic plugin panel

#### 4. Discussion

The main aim of this work is to integrate different NLP tools that could be useful for a writer trying to simplify a text in order to obtain its easy-to-read version. Tools like a dictionary of synonyms or a syntactic parser are useful in that process, but potentially difficult to use. One good example is that of an expert trying to use WordNet for synonym substitution. Although there are user-friendly interfaces to access WordNet, they are standalone applications that should be consulted externally. That is to say, if the editor finds a difficult word in the text, they have to open the WordNet application, search for the word (usually specifying if it is a noun, a verb, or an adjective), and then manually apply substitution. In order to do that several times, they would have to use the *Replace* option of the text editor. Using our lexical plugin, on the other hand, all instances of the word are highlighted, and it is easy to navigate through the text substituting the words with their synonyms. The plugin also allows the use of different synonyms in order to avoid repetition. In the case of the syntactic



simplification, our system is also useful as it allows the user to see all the conjunctions in the text and to separate complex sentences with a few simple clicks.

Even though we have here used only a limited number of tools, there are plenty more of them in the wide field of NLP, and they are not accessible to text simplification experts. In terms of lexical information, synonym search can be expanded so as to include adjectives and different verb tenses. There are also many syntactic simplifications that could be added as new tools: transformation of passive voice into active, deletion of redundant circumstantial complements, etc. Our system allows us to easily introduce new plugins that would be in charge of any new types of simplification. In addition to that, the features that are already implemented are carefully isolated so they could also be integrated in other text editors with minimum effort.

The presented tools have been published in SourceForge<sup>f,g</sup> as multiplatform free software to be freely used and improved by anyone interested. In addition to that, we are working on the publication of our plugins in the jEdit Plugin Central so that any jEdit user could benefit from them.

## 5. Conclusions and future work

We have designed and implemented a preliminary application that enables the integration of different NLP tools within an already existing text editor. It includes the components that allow for the automation of the simplification process in the future, based on simplification operations applied by the user and saved in a log. The current implementation can be easily extended by including additional simplification components in the future.

The implemented simplification features can still be improved. In the case of the lexical plugin, when a word is consulted in WordNet only the first synset is retrieved as it is the most common one. However, a word sense disambiguation algorithm could be applied to the text to retrieve the most adequate synonyms. The syntactic plugin can also be improved by including treatment of other types of complex sentences in addition to the ones that are currently treated by the system.

The information from the log can be used for a database that stores applied simplification operations, which can be suggested for similar words or phrases in texts other than the one where the change initially occurred. In addition to that, it can help understand how experts simplify texts, what the most common transformations are, whether they are performed in a specific order, etc.

The presented tool could be easily improved to work with other languages. The connection with jEdit is independent of language, although new NLP tools for the intended language would be required: natural language processors, dictionaries, etc.

Text simplification is a difficult task whose many aspects must be taken into account – from the final user to the type of text that is being simplified. As part of our future work, we intend to conduct evaluation with simplification experts in order to analyse the usability and the quality of our application.

## Acknowledgements

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

---

<sup>f</sup> Synonyms plugin: <http://sourceforge.net/projects/synonymsp/> [05-09-13]

<sup>g</sup> Split plugin: <http://sourceforge.net/projects/splitp/> [05-09-13]

## References

- [1] Freyhoff, G., G. Hess, L. Kerr, E. Menzel, B. Tronbacke and K. Van Der Veken. 1998. Make it Simple: European Guidelines for the Production of Easy-to-Read Information. ILSMH-EA.
- [2] Misako, N., G. S. Nielsen and B. Tronbacke. 1997. Guidelines for easy-to-read materials. Technical report, IFLA/Library Services to People with Special Needs Section.
- [3] Aluísio, S.M. and C. Gasperin. 2010. Fostering Digital Inclusion and Accessibility: the PorSimples Project for Simplification of Portuguese Texts. In Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas. ACL.
- [4] Carroll, J., G. Minnen, Y. Canning, S. Devlin and J. Tait. 1998. Practical Simplification of English Newspaper Text to Assist Aphasic Readers. In AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology.
- [5] Miller, G.A., R. Beckwith, C. Fellbaum, D. Gross and K.J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database. *Int J Lexicography*, 3(4):235-244.
- [6] Quinlan, Philip. 1992. The Oxford Psycholinguistic Database. Oxford University Press, U.K.
- [7] Caseli, H.M., T.F. Pereira, L. Specia, A. Thiago, S. Pardo, C. Gasperin and S.M. Aluísio. 2009. Building a Brazilian Portuguese Parallel Corpus of Original and Simplified Texts. In 10th Conference on Intelligent Text Processing and Computational Linguistics.
- [8] Saggion, H., E. Gómez-Martínez, E. Etayo, A. Anula and L. Bourg. 2011. Text Simplification in Simplex: Making Text More Accessible. *Procesamiento del Lenguaje Natural*, 47.
- [9] Alusío, S.M., Specia, L., Pardo, S., Maziero, E.G. and Fortes, R.P.M. 2008. Towards Brazilian Portuguese Automatic Text Simplification Systems. In Proceedings of the Eighth ACM symposium on Document Engineering.
- [10] Candido, A. Jr., Maziero, E.G., Gasperin, C., Pardo, S., Specia, L. and Aluísio, S.M. 2009. Supporting the Adaptation of Texts for Poor Literacy Readers: a Text Simplification Editor for Brazilian Portuguese. In Proceedings of EdAppsNLP, ACL.
- [11] Al-Subaihín, A.A., Al-Khalifa, H.S. 2011. Al-baseet: A Proposed Simplification Authoring Tool for the Arabic Language. In Proceedings of ICCIT.
- [12] Rybing, J., Smith, D. and Sivervarg, A. 2010. Towards a Rule Based System for Automatic Simplification of Texts. In Proceedings of SLTC.
- [13] Siddharthan, A. 2006. Syntactic Simplification and Text Cohesion. *Research on Language & Computation*, 4(1): 77-109.
- [14] Aranzabe, M., Díaz de Ilaraza, A., González, I. 2012. First Approach to Automatic Text Simplification in Basque. In Proceedings of NLP4ITA. LREC.
- [15] Drndarevic, B. and H. Saggion. 2012. Reducing Text Complexity through Automatic Lexical Simplification: an Empirical Study for Spanish. *Procesamiento del Lenguaje Natural*, 49.
- [16] Drndarevic, B. and H. Saggion. 2012. Towards Automatic Lexical Simplification in Spanish: An Empirical Study. In Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations, ACL.
- [17] Seco, N. LightWeight WordNet. 2005. <http://eden.dei.uc.pt/~nseco/lwwn.tar.gz> [05-09-13].