Towards an on-demand Simple Portuguese Wikipedia

Arnaldo Candido Junior
Institute of Mathematics and Computer Sciences
University of São Paulo
arnaldoc at icmc.usp.br

Ann Copestake
Computer Laboratory
University of Cambridge
Ann.Copestake at cl.cam.ac.uk

Lucia Specia
Research Group in Computational Linguistics
University of Wolverhampton
l.specia at wlv.ac.uk

Sandra Maria Aluísio
Institute of Mathematics and Computer Sciences
University of São Paulo
sandra at icmc.usp.br

Abstract

The *Simple English Wikipedia* provides a simplified version of Wikipedia's English articles for readers with special needs. However, there are fewer efforts to make information in Wikipedia in other languages accessible to a large audience. This work proposes the use of a syntactic simplification engine with high precision rules to automatically generate a Simple Portuguese Wikipedia on demand, based on user interactions with the main Portuguese Wikipedia. Our estimates indicated that a human can simplify about 28,000 occurrences of analysed patterns per million words, while our system can correctly simplify 22,200 occurrences, with estimated f-measure 77.2%.

1 Introduction

The *Simple English Wikipedia*[1] is an effort to make information in Wikipedia[2] accessible for less competent readers of English by using simple words and grammar. Examples of intended users include children and readers with special needs, such as users with learning disabilities and learners of English as a second language.

Simple English (or Plain English), used in this version of Wikipedia, is a result from the Plain English movement that occurred in Britain and the United States in the late 1970's as a reaction to the unclear language used in government and business forms and documents. Some recommendations on how to write and organize information in Plain Language (the set of guidelines to write simplified texts) are related to both syntax and lexical levels: use short sentences; avoid hidden verbs; use active voice; use concrete, short, simple words.

A number of resources, such as lists of common words[3], are available for the English language to help users write in Simple English. These include lexical resources like the MRC Psycholinguistic Database[4] which helps identify difficult words using psycholinguistic measures. However, resources as such do not exist for Portuguese. An exception is a small list of simple words compiled as part of the PorSimples project (Aluisio et al., 2008).

Although the guidelines from the Plain Language can in principle be applied for many languages and text genres, for Portuguese there are very few efforts using Plain Language to make information accessible to a large audience. To the best of our knowledge, the solution offered by *Portugues Claro*[5] to help organizations produce European Portuguese (EP) documents in simple language is the only commercial option in such a direction. For Brazilian Portuguese (BP), a Brazilian Law (10098/2000) tries to ensure that content in e-Gov sites and services is written in simple and direct language in order to remove barriers in communication and to ensure citizens' rights to information and communication access. However, as it has been shown in Martins and Filgueiras (2007), content in such websites still needs considerable rewriting to follow the Plain Language guidelines.

A few efforts from the research community have recently resulted in natural language processing

1 http://simple.wikipedia.org/
2 http://www.wikipedia.org/
3 http://simple.wiktionary.org/
4 http://www2.let.vu.nl/resources/elw/resource/mrc.html
5 http://www.portuguesclaro.pt/

Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pages 137–147,
Edinburgh, Scotland, UK, July 30, 2011.

systems to simplify and make Portuguese language clearer. ReEscreve (Barreiro and Cabral, 2009) is a multi-purpose paraphraser that helps users to simplify their EP texts by reducing its ambiguity, number of words and complexity. The current linguistic phenomena paraphrased are support verb constructions, which are replaced by stylistic variants. In the case of BP, the lack of simplification systems led to development of PorSimples project (Aluísio and Gasperin, 2010). This project uses simplification in different linguistic levels to provide simplified text to poor literacy readers.

For English, automatic text simplification has been exploited for helping readers with poor literacy (Max, 2006) and readers with other special needs, such as aphasic people (Devlin and Unthank, 2006; Carroll et al. 1999). It has also been used in bilingual education (Petersen, 2007) and for improving the accuracy of Natural Language Processing (NLP) tasks (Klebanov et al., 2004; Vickrey and Koller, 2008).

Given the general scarcity of human resources to manually simplify large content repositories such as Wikipedia, simplifying texts automatically can be the only feasible option. The Portuguese Wikipedia, for example, is the tenth largest Wikipedia (as of May 2011), with 683,215 articles and approximately 860,242 contributors[6].

In this paper we propose a new rule-based syntactic simplification system to create a Simple Portuguese Wikipedia on demand, based on user interactions with the main Portuguese Wikipedia. We use a simplification engine to change passive into active voice and to break down and change the syntax of subordinate clauses. We focus on these operations because they are more difficult to process by readers with learning disabilities as compared to others such as coordination and complex noun phrases (Abedi et al., 2011; Jones et al., 2006; Chappell, 1985). User interaction with Wikipedia can be performed by a system like the Facilita[7] (Watanabe et al., 2009), a browser plug-in developed in the PorSimples project to allow automatic adaptation (summarization and syntactic simplification) of any web page in BP.

This paper is organized as follows. Section 2 presents related work on syntactic simplification. Section 3 presents the methodology to build and evaluate the simplification engine for BP. Section 4 presents the results of the engine evaluation. Section 5 presents an analysis on simplification issues and discusses possible improvements. Section 6 contains some final remarks.

2 Related work

Given the dependence of syntactic simplification on linguistic information, successful approaches are mostly based on rule-based systems. Approaches using operations learned from corpus have not shown to be able to perform complex operations such the splitting of sentences with relative clauses (Chandrasekar and Srinivas, 1997; Daelemans et al., 2004; Specia, 2010). On the other hand. the use of machine learning techniques to predict when to simplify a sentence, i.e. learning the properties of language that distinguish simple from normal texts, has achieved relative success (Napoles and Dredze, 2010). Therefore, most work on syntactic simplification still relies on rule-based systems to simplify a set of syntactic constructions. This is also the approach we follow in this paper. In what follows we review some relevant and work on syntactic simplification.

The seminal work of Chandrasekar and Srinivas (1997) investigated the induction of syntactic rules from a corpus annotated with part-of-speech tags augmented by agreement and subcategorization information. They extracted syntactic correspondences and generated rules aiming to speed up parsing and improving its accuracy, but not working on naturally occurring texts. Daelemans et al. (2004) compared both machine learning and rule-based approaches for the automatic generation of TV subtitles for hearing-impaired people. In their machine learning approach, a simplification model is learned from parallel corpora with TV programme transcripts and the associated subtitles. Their method used a memory-based learner and features such as words, lemmas, POS tags, chunk tags, relation tags and proper name tags, among others features (30 in total). However, this approach did not perform as well as the authors expected, making errors like removing sentence subjects or deleting a part of a multi-word unit. More recently, Specia (2010) presented a new approach for text simplification, based on the framework of Statistical Machine

6 http://meta.wikimedia.org/wiki/List_of_Wikipedias#Grand Total

7 http://nilc.icmc.usp.br/porsimples/facilita/

Translation. Although the results are promising for lexical simplification, syntactic rewriting was not captured by the model to address long-distance operations, since syntactic information was not included into the framework.

Inui et al. (2003) proposed a rule-based system for text simplification aimed at deaf people. Using about one thousand manually created rules, the authors generate several paraphrases for each sentence and train a classifier to select the simpler ones. Promising results were obtained, although different types of errors on the paraphrase generation are encountered, such as problems with verb conjugation and regency. Our work aims at making Portuguese Wikipedia information accessible to a large audience and instead of generating several possible outputs we generate only one based on rules taken from a manual of simplification for BP.

Siddharthan (2006) proposed a syntactic simplification architecture that relies on shallow parsing. The general goal of the architecture is to make texts more accessible to a broader audience instead of targeting any particular application. The system simplifies apposition, relative clauses, coordination and subordination. Our method, on the other hand, relies on deep parsing (Bick, 2000) and focuses on changing passive to active voice and changing the syntax of relative clauses and subordinate sentences.

Max (2006) applied text simplification in the writing process by embedding the simplifier into a word processor. Although this system ensures accurate output, it requires manual choices. The suggested simplifications are ranked by a score of syntactic complexity and potential change of meaning. The writer then chooses their preferred simplification. Our method, on the other hand, offers the user only one simplification since it uses several rules to better capture each complex phenomenon.

Inspired by Siddharthan (2006), Jonnalagadda and Gonzalez (2009) present an approach to syntactic simplification addressing also the problem of accurately determining the grammatical correctness of the simplified sentences. They propose the combination of the number of null links and disjunct cost (the level of inappropriateness, caused by using less frequent rules in the linkage) from the cost vector returned by a Link Grammar[8] parser. Their motivation is to improve the performance of systems for extracting Protein-Protein Interactions automatically from biomedical articles by automatically simplifying sentences. Besides treating the syntactic phenomena described in Siddharthan (2006), they remove describing phrases occurring at the beginning of the sentences, like "These results suggest that" and "As reported previously". While they focus on the scientific genre, our work is focused on the encyclopedic genre.

In order to obtain a text easier to understand by children, De Belder and Moens (2010) use the Stanford parser[9] to select the following phenomena to syntactically simplify the sentences: appositions, relative clauses, prefix subordination and infix subordination and coordination. After sentence splitting, they try to apply the simplification rules again to both of the new sentences. However, they conclude that with the set of simplification rules used, it was not possible to reduce the reading difficulty for children and foresee the use of other techniques for this purpose, such as summarization and elaborations for difficult words.

3 Simplification engine

3.1 Engine development

The development of a syntactic simplification engine for a specific task and audience can be divided into five distinct phases: (a) target audience analysis; (b) review of complex syntactic phenomena for such an audience; (c) formulation of simplification guidelines; (d) refinement of rules based on evidence from corpora; and (e) programming and evaluation of rules.

In this paper we focus on the last two phases. We use the simplification guidelines from the PorSimples project, but these are based on grammar studies and corpora analysis for a different text genre (news). Therefore additional corpora evidence proved to be necessary. This resulted in the further refinement of the rules, covering different cases for each syntactic phenomenon.

The Simplification engine relies on the output of the Palavras Parser (Bick, 2000) to perform constituent tree transformations (for example, tree

8 http://www.abisource.com/projects/link-grammar/
9 http://nlp.stanford.edu/software/lex-parser.shtml

splitting). Each node of a sentence tree is fed (breadth-first order) to the simplification algorithms, which can simplify the node (and its sub-tree) or skip it when the node does not meet the simplification prerequisites. Breadth-first order is chosen because several operations affect the root of a (sub)tree, while none of them affect leaves.

A development corpus containing examples of cases analysed for each syntactic phenomenon is used to test and refine the rules. The current version of the corpus has 156 sentences extracted from news text. The corpus includes negative and positive examples for each rule. Negative examples should not be simplified. They were inserted into the corpus to avoid unnecessary simplifications. Each rule is first tested against its own positive and negative examples. This test is called *local test*. After reaching a good precision on the local test, the rule is then tested against all the sentences in the corpus, *global test*. In the current corpus, the global test identified sentences correctly simplified by at least one rule (66%), sentences incorrectly simplified due to major errors in parsing/rules (7%) (ungrammatical sentences) and non-simplified sentences (27%). The last includes mainly negative examples, but also includes sentences not selected due to parsing errors, sentences from cases not yet implemented, and sentences from cases ignored due to ambiguity.

3.2 Passive voice

The default case for dealing with passive voice in our simplification engine is illustrated by the pair of original-simplified sentences in example[10] (1). Sentences belonging to this case have a non-pronominal subject and a passive agent. Also, the predicator has two verbs, the verb *to be* followed by a verb in the past participle tense. The simplification consists in reordering the sentence components, turning the agent into subject (removing the *by* preposition), turning the subject into direct object and adjusting the predicator by removing the verb *to be* and re-inflecting the main verb. The new tense of the main verb is the same as the one of the *to be* verb and its number is defined according to the new subject.

O: As[The] transferências[transfers] foram[were:plural] feitas[made] pela[by the] empresa[company]. (1)
S: A[The] empresa[company] fez[made:sing] as[the] transferências[transfers].

Other correctly processed cases vary according the number of verbs (three or four), special subjects, and special agents. For cases comprising three or four verbs, the simplification rule must re-inflect[11] two verbs (2) (one of them should agree with the subject and the other receives its tense from the verb *to be*). There are two cases of special subjects. In the first case, a hidden subject is turned into a pronominal direct object (3). In the second case, a pronominal subject must be transformed to oblique case pronoun and then to direct object. Special agents also represent two cases. In the first one, oblique case pronouns must be transformed before turning the agent into the subject. In the second case (4), a non-existent agent is turned into an undetermined subject (represented here by "they").

O: A[The] porta[door] deveria[should] ter[have] sido[been] trancada[locked:fem] por[by] John. (2)
S: John deveria[should] ter[have] trancado[locked:masc] a[the] porta[door].

O: [I] fui[was] encarregado[entrusted] por[by] minha[my] família[family]. (3)
S: Minha[My] família[family] encarregou[entrusted] me[me].

O: O[The] ladrão[thief] foi[was] pego[caught]. (4)
S: [They] pegaram[caught] o[the] ladrão[thief].

Two cases are not processed because they are already considered easy enough: the syndetic voice and passive in non-root sentences. In those cases, the proposed simplification is generally less understandable than the original sentence. Sentences with split predicator (as in "the politician was very criticized by his electors") are not processed for the time being, but should be incorporated in the pipeline in the future.

Table 1 presents the algorithm used to process the default case rule and verb case rules. Simplification rules are applied against all nodes in constituent tree, one node at a time, using breadth-first traversing.

10 Literal translations from Portuguese result in some sentences appearing ungrammatical in English.

11 Some reinflections may not be visible on example translation.

Step	Description
1	Validate these prerequisites or give up:
1.1	Node must be root
1.2	Predictor must have an inflection of auxiliary verb *to be*
1.3	Main verb has to be in past participle
2	Transform subject into direct object
3	Fix the predicator
3.1	If main verb is finite then: main verb gets mode and tense from *to be* main verb gets person according to agent
3.2	Else: main verb gets mode and tense from verb *to be* finite verb gets person according to agent
3.3	Remove verb *to be*
4	Transform passive agent into a new subject

Table 1: Algorithm for default and verb cases

3.3 Subordination

Types of subordinate clauses are presented in Table 2. Two clauses are not processed: comparative and proportional. Comparative and proportional clauses will be addressed in future work.

id	Clause type	Processed
d	Relative Restrictive	✓
e	Relative Non-restrictive	✓
f	Reason	✓
g	Comparative	
h	Concessive	✓
i	Conditional	✓
j	Result	✓
k	Confirmative	✓
l	Final Purpose	✓
m	Time	✓
w	Proportional	

Table 2: Subordinate clauses

Specific rules are used for groups of related subordinate cases. At least one of two operations can be found in all rules: component reordering and sentence splitting. Below, letter codes are used to describe rules involving these two and other common operations:

A	additional processing
M	splitting-order main-subordinate
P	Also processes non-clause phrases and/or non-finite clauses
R	component reordering
S	splitting-order subordinate-main
c	clone subject or turn object of a clause into subject in another if it is necessary
d	marker deletion
m	marker replacement
v	verb reinflection
[na]	not simplified due ambiguity
[nf]	not simplified, future case
[np]	not simplified due parsing problems
2...8	covered cases (when more than one applies)

Table 3 presents the marker information. They are used to select sentences for simplification, and several of them are replaced by easier markers. Cases themselves are not detailed since they are too numerous (more than 40 distinct cases). Operation codes used for each marker are described in column "Op". It is important to notice that multi-lexeme markers also face ambiguities due to co-occurrence of its component lexemes[12]. The list does not cover all possible cases, since there may be additional cases not seen in the corpus. As relative clauses (*d* and *e*) require almost the same processing, they are grouped together.

Several clauses require additional processing. For example, some conditional clauses require negating the main clause. Other examples include noun phrases replacing clause markers and clause reordering, both for relative clauses, as showed in (5). The marker *cujo* (*whose*) in the example can refer to *Northbridge* or to *the building*. Additional processing is performed to try to solve this anaphora[13], mostly using number agreement between the each possible co-referent and the main verb in the subordinate clause. The simplification engine can give up in ambiguous cases (focusing on precision) or elect a coreferent (focusing on recall), depending on the number of possible coreferents and on a confidence threshold parameter, which was not used in this paper.

O: Ele[He] deve[should] visitar[visit] o[the] prédio[building] em[in] Northbridge cujo[whose] desabamento[landslide] matou[killed] 16 pessoas[people].
S: Ele[He] deve[should] visitar[visit] o[the] prédio[building] em[in] Northbridge. O[The] desabamento[landslide] do[of the] prédio[building] em[in] Northbridge matou[killed] 16 pessoas[people]. (5)

12 For example, words "de", "sorte" and "que" can be adjacent to each other without the meaning of "de sorte que" marker ("so that").

13 We opted to solve this kind of anaphora instead of using pronoun insertion in order to facilitate the reading of the text.

id	Marker	Op	id	Marker	Op	id	Markers	Op
de	que [that/which]	8MRAdv	h	se bem que [albeit]	Mmv	j	tanto … que [so … that]	[nf]
de	o qual [which]*	8MRAdv	h	ainda que [even if]	2Mm	j	tal … que [such … that]	[nf]
de	como [as]	[na]	h	mesmo que [even if]	2Mm	j	tamanho … que [so … that]*	[nf]
de	onde [where]	[nf]	h	nem que [even if]	2Mm	k	conforme [as/according]	3PRAcm
de	quando [when]	[na]	h	por mais que [whatever]	2Mm	k	consoante [as/according]	3PRAcm
de	quem [who/whom]	[nf]	h	mas [but]	[np]	k	segundo [as/according]	3PRAcm
de	quanto [how much]	[nf]	i	contanto que [provided that]	2Rmv	k	como [as]	[na]
de	cujo [whose]*	MAd	i	caso [case]	2Rmv	l	a fim de [in order to]	2PMcm
de	o que [what/which]	Sd	i	se [if/whether]	2Rmv	l	a fim de que [in order that]	2PMcm
f	já que [since]	Scm	i	a menos que [unless]	2RAmv	l	para que [so that]	2PMcm
f	porquanto [in view of]	Scm	i	a não ser que [unless]	2RAmv	l	porque [because]	[na]
f	uma vez que [since]	Scm	i	exceto se [unless]	2RAmv	m	assim que [as soon as]	5PMAcvr
f	visto que [since]	Scm	i	salvo se [unless]	2RAmv	m	depois de [after]	5PMAcvr
f	como [for]	[na]	i	antes que [before]	Rmv	m	depois que [after]	5PMAcvr
f	porque [because]	[na]	i	sem que [without]	Rmv	m	logo que [once]	5PMAcvr
f	posto que [since]	[na]	i	desde que [since]	RAmv	m	antes que [before]	PSAcvr
f	visto como [seen as]	[na]	j	de forma que [so]	5Mmv	m	apenas [only]	[na]
f	pois que [since]	[nf]	j	de modo que [so]	5Mmv	m	até que [until]	[na]
h	apesar de que [although]	Mmv	j	de sorte que [so that]	5Mmv	m	desde que [since]	[na]
h	apesar que [despite]	Mmv	j	tamanho que [so that]*	5Mmv	m	cada vez que [every time]	[nf]
h	conquanto [although]	Mmv	j	tal que [such that]	5Mmv	m	sempre que [whenever]	[nf]
h	embora [albeit]	Mmv	j	tanto que [so that] (1)*	[na]	m	enquanto [while]	[nf]
h	posto que [since]	Mmv	j	tanto que [so that] (2)	[na]	m	mal [just]	[na]
h	por muito que [although]	Mmv	j	tão … que [so … that]	[nf]	m	quando [when]	[na]

* gender and/or number variation

Table 3: Marker processing

3.4 Evaluation in the development corpus

Figure 1 provides statistics from the of processing all identified cases in the development corpus. These statistics cover number of cases rather than the number of sentences containing cases. The cases "incorrect selection" and "incorrect simplification" affect precision by generating ungrammatical sentences. The former refers to sentences that should not be selected for the simplification process, while the latter refers to sentences correctly selected but wrongly simplified. There are three categories affecting recall, classified according to their priority in the simplification engine. *Pending* cases are considered to be representative, with higher priority. *Possible* cases are considered to be unrepresentative. Having less priority, they can be handled in future versions of the engine. Finally, *Skipped* cases will not be implemented, mainly because of ambiguity, but also due to low representativeness. It is possible to observe that categories reducing precision (incorrect selection and simplification) represent a smaller number of cases (5%) than categories reducing recall (45%). It is worth noticing that our approach focus on precision in order to make the simplification as automatic as possible, minimizing the need for human interaction.

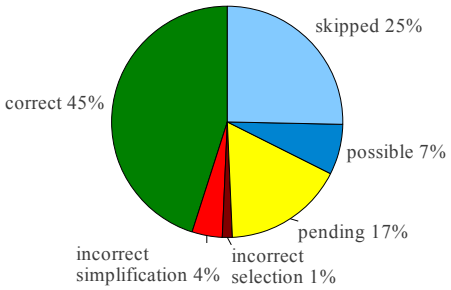

Figure 1: Performance on the development corpus

There are some important remarks regarding the development corpus used during the programming phase. First, some cases are not representative, therefore the results are expected to vary significantly in real texts. Second, a few cases are not orthogonal: i.e., there are sentences that can be classified in more than one case. Third, several errors refer to sub-cases of cases being mostly correctly processed, which are expected to occur less frequently. Fourth, incorrect parsed sentences were not take in account in this phase. Although there may exist other cases not identified yet, it is plausible to estimate that only 5% of known cases are affecting the precision negatively.

4 Engine evaluation

4.1 Evaluation patterns

The evaluation was performed on a sample of sentences extracted from Wikipedia's texts using lexical patterns. These patterns allows to filter the texts, extracting only relevant sentences for precision and recall evaluation. They were created to cover both positive and negative sentences. They are applied before parsing or Part of Speech (PoS) analysis. For passive voice detection, the pattern is defined as a sequence of two or more possible verbs (no PoS in use) in which at least one of them could be an inflection of verb to be. For subordination detection, the pattern is equivalent to the discourse markers associated with each subordination type, as shown in Table 3.

The patterns were applied against featured articles appearing in Wikipedia's front page in 2010 and 2011, including featured articles planned to be featured, but not featured yet. A maximum of 30 sentences resulting from each pattern matching were then submitted to the simplification engine. Table 4 presents statistics from featured articles.

texts	165
sentences	83,656
words	1,226,880
applied patterns	57,735
matched sentences	31,080

Table 4: Wikipedia's featured articles (2010/2011)

The number of applied patterns represents both patterns to be simplified (s-patterns) and patterns not to be simplified (n-patterns). N-patterns represent both non-processable patterns due to high ambiguity (a-patterns) and pattern extraction false negatives. We observed a few, but very frequent, ambiguous patterns introducing noise, particularly *se* and *como*. In fact, these two markers are so noisy that we were not be able to provide good estimations on their true positives distribution given the 30 sentences limit per pattern. Similarly to the number of applied patterns, the number of matched sentences correspond to both sentences to be simplified and not to be simplified.

Table 5 presents additional statistics about characters, words and sentences calculated in a sample of 32 articles where the 12 domains of the Portuguese Wikipedia are balanced. The number of automatic simplified sentence is also presented. In Table 5, *simple words* refers to percentage of words which are listed on our simple word list, supposed to be common to youngsters, extracted from the dictionary described in (Biderman, 2005), containing 5,900 entries. Figure 2 presents clause distribution per sentence in the balanced sample. *Zero clauses* refers to titles, references, figure labels, and other pieces of text without a verb. We observed 60% of multi-clause sentences in the sample.

characters per word	5.22
words per sentence	21.17
words per text	8,476
simple words	75.52%
sentences per text	400.34
passive voice	15.11%
total sentences	13,091
simplified sentences	16,71%

Table 5: Statistics from the balanced text sample

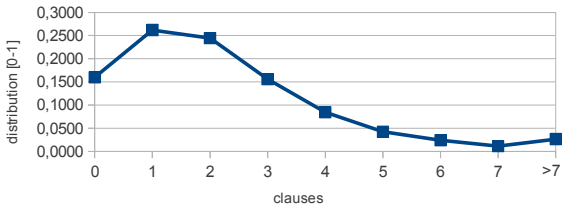

Figure 2: Clauses per sentence in the sample

4.2 Simplification analysis

We manually analysed and annotated all sentences in our samples. These samples were used to estimate several statistics, including the number of patterns per million words, the system precision and recall and the noise rate. We opted for analysing simplified patterns per million words instead of per simplified sentences. First, because an analysis based on sentences can be misleading, since there are cases of long coordinations with many patterns, as well as succinct sentences with no patterns. Moreover, one incorrectly simplified marker in a sentence could hide useful statistics of correctly simplified patterns and even of other incorrectly simplified patterns.

The samples are composed by s-patterns and n-patterns (including a-patterns). In total 1,243 patterns were annotated. Table 6 presents pattern estimates per million words.

Total patterns	70,834
Human s-patterns	33,906
Selection s-patterns	27,714
Perfect parser s-patterns	23,969
Obtained s-patterns	22,222

Table 6: Patterns per million words

Total patterns refers to the expected occurrences of s-patterns and n-patterns in a corpus of one million words. This is the only information extracted from the full corpus, while the remaining figures are estimates from the sample corpus.

Human s-patterns is an estimate of the number patterns that a human could simplify in the corpus. Unlike other s-pattern estimates, a-patterns are included, since a human can disambiguate them. In other words, this is the total of positive patterns. The estimate does not include very rare (sample size equals to zero) or very noisy markers (patterns presenting 30 noisy sentences in its sample).

Selection s-patterns are an estimate of the number of patterns correctly selected for simplification, regardless of whether the pattern simplification is correct or incorrect. Precision and recall derived from this measure (Table 7) consider incorrectly simplified patterns, and do not include patterns with parsing problems. Its purpose is to evaluate how well the selection for simplification is performed. Rare or noisy patterns, whose human s-patterns per sample is lower than 7, are not included.

Perfect parser s-patterns is an estimate very similar to selection s-patterns, but considering only correctly simplified patterns. As in selection s-patterns, incorrect parsed sentences are not included in calculations. This is useful to analyse incorrect simplifications due to simplification rule problems, ignoring errors originating from parsing.

Finally, *obtained s-patterns* refers to the estimate of correct simplified patterns, similar to perfect parser s-patterns, but including simplification problems caused by parsing. This estimate represents the real performance to be expected from the system on Wikipedia's texts.

It is important to note that the real numbers of *selection s-patterns*, *perfect s-patterns* and *obtained s-patterns* is expected to be bigger than the estimates, since noisy and rare pattern could not used be used in calculations (due the threshold of 7 human s-patterns per sample). The data presented on Table 6 is calculated using estimated local precisions for each pattern. Table 7 presents global precision, recall and f-measure related to *selection, perfect parser* and *obtained s-patterns*. The real values of the estimates are expected to variate up to +/- 2.48% .

Measures	**Precision**	**Recall**	**F-measure**
Selection	99.05%	82.24%	89.86%
Perfect parser	85.66%	82.24%	83.92%
Obtained	79.42%	75.09%	77.20%

Table 7: Global estimated measures

Although the precision of the selection seems to be impressive, this result is expected, since our approach focus on the processing of mostly unambiguous markers, with sufficient syntactic information. It is also due to the the threshold of 7 human s-patterns and the fact that a-patterns are not included. Due to these two restrictions, only approximately 31.5% of unique patterns could be used for the calculations in Table 7. Interestingly, these unique patterns correspond to 82.5% of the total estimated human s-patterns. The majority of the 17.5% remaining s-patterns refers to patterns too noisy to be analysed and to a-patterns (not processed due ambiguity), and also others n-patterns which presented a low representativeness in the corpus. The results indicate good performance in rule formulation, covering the most important (and non-ambiguous) markers, which is also confirmed by the ratio between both selection s-patterns and human s-patterns previously presented on Table 6.

An alternative analysis, including a-patterns, lowers recall and f-measure, but not precision (our focus in this work). In this case, recall drops from 75.09% to 62.18%, while f-measure drops from 77.20% to 70.18%.

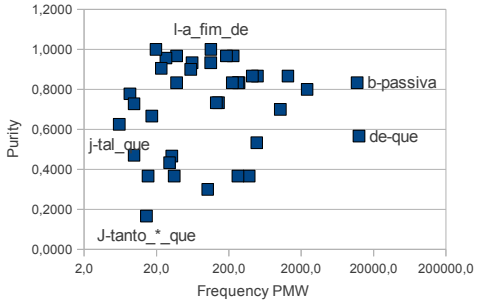

Figure 3: Pattern distribution

Figure 3 presents the distribution of patterns according to their frequency per million words and their purity (1 - noisy rate). This data is useful to

identify most frequent patterns (such as passive voice in *b-passiva*) and patterns with medium to high frequency, which are easy to process (not ambiguous), such as *l-a_fim_de*.

5 Issues on simplification quality

This analysis aims at identifying factors affecting the quality of simplifications considered as correct. Hence, factors affecting the overall simplified text quality are also presented. In contrast, the quantitative analysis presented on Section 4.2 covered the ratio between incorrect and correct simplifications.

Three cases of clause disposition were identified as important factors affecting the simplified sentence readability. These cases are presented using the following notation: clauses are represented in uppercase letters; clause concatenation represents coordination; parentheses represent subordination; c_1 and c_2 represent clause/sentence connectors (including markers); the entailment operator (→) represents the simplification rule transforming clauses.

- "A(B(c_1 C)) → A(B). c_2 C": the **vertical case**. In this scenario it is more natural to read c_2 as connecting *C* to the main clause *A*, while c_1 connects *C* to *B*, as seen in (6). This is still acceptable for several sentences analysed, but we are considering to simplify only level 2 clauses in the future, splitting *C* from B only if another rule splits A and B first.
- "A(B)CD → ACD. c_1 B": the **horizontal case**. In this scenario, c_1 correctly connects *A* and *B*, but long coordinations following A can impact negatively on text reading, since the target audience may forget about *A* when starting to read *B*. In this scenario, coordination compromise subordination simplification, showing the importance of simplifying coordination as well, even though they are considered easier to read than subordination.
- **Mixed case**: this scenario combines the potential problems of horizontal and vertical cases. It may occur in extremely long sentences.

Besides clause disposition factors, clause inversions can also lead to problems in sentence readability. In our current system, inversion is mainly used to produce simplified sentences in the cause-effect order or condition-action order. Reordering, despite using more natural orders, can transform anaphors into cataphors. A good anaphora resolution system would be necessary to avoid this issue. Another problem is moving sentence connectors as in "A. c_1 BC. → A. B. c_2 c_1 C", while "A. c_1 B. c_2 C" is more natural (maintaining c_1 position).

O: Ela[She] dissertou[talked] sobre[about] como[how] motivar[to motive] o[the] grupo[group] de_modo_que[so that] seu[their] desempenho[performance] melhore[improves]
S: [He/She] dissertou[talked] sobre[about] como[how] motivar[to motive] o[the] grupo[group]. Thus, seu[their] desempenho[performance] melhore[improves] (6)

We have observed some errors in sentence parsing, related to clause attachment, generating truncated ungrammatical text. As a result, a badly simplified key sentence can compromise the text readability more than several correctly simplified sentences can improve it, reinforcing the importance of precision rather than recall in automated text simplification.

Experienced readers analysed the simplified versions of the articles and considered them easier to read than the original ones in most cases, despite simplification errors. Particularly, the readers considered that the readability would improve significantly if cataphor and horizontal problems were addressed. Evaluating the simplifications with readers from the target audience is left as a future work, after improvements in the identified issues.

6 Conclusions

We have presented a simplification engine to process texts from the Portuguese Wikipedia. Our quantitative analysis indicated a good precision (79.42%), and reasonable number of correct simplifications per million words (22,222). Although our focus was on the encyclopedic genre evaluation, the proposed system can be used in other genres as well.

Acknowledgements
We thank FAPESP (p. 2008/08963-4) and CNPq (p. 201407/2010-8) for supporting this work.

References

J. Abedi, S. Leon, J. Kao, R. Bayley, N. Ewers, J. Herman and K. Mundhenk. 2011. Accessible Reading Assessments for Students with Disabilities: The Role of Cognitive, Grammatical, Lexical, and Textual/Visual Features. CRESST Report 785. National Center for Research on Evaluation, Standards, and Student Testing, University of California, Los Angeles.

S. M. Aluísio, C. Gasperin. 2010. Fostering Digital Inclusion and Accessibility: The PorSimples project for Simplification of Portuguese Texts. Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas. : ACL, New York, USA. v. 1. p. 46-53.

A. Barreiro, L. M. Cabral. 2009. ReEscreve: a translator-friendly multi-purpose paraphrasing software tool. The Proceedings of the Workshop Beyond Translation Memories: New Tools for Translators, The Twelfth Machine Translation Summit. Ontario, Canada, pp. 1-8.

E. Bick. 2006. The parsing system "Palavras": Automatic grammatical analysis of Portuguese in a constraint grammar framework. Thesis (PhD). University of Århus, Aarhus, Denmark.

M. T. C. Biderman. 2005. Dicionário Ilustrado de Português. Editora Ática. 1a. ed. São Paulo

J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin and J. Tait. 1999. Simplifying Text for Language-Impaired Readers,. In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 269-270.

R. Chandrasekar and B. Srinivas. 1997. Automatic Induction of Rules for Text Simplification. Knowledge-Based Systems, 10, 183-190.

G. E. Chappell. 1985. Description and assessment of language disabilities of junior high school students. In: Communication skills and classroom success: Assessment of language-learning disabled students. College- Hill Press, San Diego, pp. 207-239.

W. Daelemans, A. Hothker and E. T. K. Sang. 2004. Automatic Sentence Simplification for Subtitling in Dutch and English. In: Proceedings of the 4th Conference on Language Resources and Evaluation, Lisbon, Portugal 1045-1048.

J. De Belder and M. Moens. 2010. Text simplification for children. Proceedings of the SIGIR Workshop on Accessible Search Systems, pp.19-26.

S. Devlin and G. Unthank. 2006. Helping aphasic people process online information. In: Proceedings of the ACM SIGACCESS 2006, Conference on Computers and Accessibility. Portland, Oregon, USA , 225-226.

K. Inui, A. Fujita, T. Takahashi, R. Iida and T. Iwakura. 2003. Text Simplification for Reading Assistance: A Project Note. In the Proceedings of the Second International Workshop on Paraphrasing, 9-16.

S. Jonnalagadda and G. Gonzalez. 2009. Sentence Simplification Aids Protein-Protein Interaction Extraction. Proceedings of the 3rd International Symposium on Languages in Biology and Medicine, Short Papers, pages 109-114, Jeju Island, South Korea, 8-10 November 2009.

F. W. Jones, K. Long and W. M. L. Finlay. 2006. Assessing the reading comprehension of adults with learning disabilities. Journal of Intellectual Disability Research, 50(6), 410-418.

B. Klebanov, K. Knight and D. Marcu. 2004. Text Simplification for Information-Seeking Applications. In: On the Move to Meaningful Internet Systems. Volume 3290, Springer-Verlag, Berlin Heidelberg New York, 735-747.

S. Martins, L. Filgueiras. 2007. Métodos de Avaliação de Apreensibilidade das Informações Textuais: uma Aplicação em Sítios de Governo Eletrônico. In proceeding of Latin American Conference on Human-Computer Interaction (CLIHC 2007). Rio de Janeiro, Brazil.

A. Max. 2006. Writing for Language-impaired Readers. In: Proceedings of Seventh International Conference on Intelligent Text Processing and Computational Linguistics. Mexico City, Mexico. Berlin Heidelberg New York, Springer-Verlag, 567-570.

C. Napoles and M. Dredze. 2010. Learning simple Wikipedia: a cogitation in ascertaining abecedarian language. In the Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids (CL&W '10), 42-50.

S. E. Petersen. 2007. Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education. PhD thesis. University of Washington.

A. Siddharthan. 2006. Syntactic simplification and text cohesion. Research on Language & Computation, 4(1):77-109.

L. Specia. 2010. Translating from Complex to Simplified Sentences. 9th International Conference

on Computational Processing of the Portuguese Language. Lecture Notes in Artificial Intelligence, Vol. 6001, Springer, pp. 30-39.

D. Vickrey and D. Koller. 2008. Sentence Simplification for Semantic Role Labelling. In: Proceedings of the ACL-HLT. 344-352.

W. M. Watanabe, A. Candido Jr, V. R. Uzeda, R. P. M. Fortes, T. A. S. Pardo and S. M. Aluísio. 2009. Facilita: Reading Assistance for Low-literacy Readers. In: ACM International Conference on Design of Communication (SIGDOC 2009), volume 1, Bloomington, US, 29-36.