An evaluation of syntactic simplification rules for people with autism

Richard Evans, Constantin Orăsan and Iustin Dornescu
Research Institute in Information and Language Processing
University of Wolverhampton
United Kingdom
{R.J.Evans, C.Orasan, I.Dornescu2}@wlv.ac.uk

Abstract

Syntactically complex sentences constitute an obstacle for some people with Autistic Spectrum Disorders. This paper evaluates a set of simplification rules specifically designed for tackling complex and compound sentences. In total, 127 different rules were developed for the rewriting of complex sentences and 56 for the rewriting of compound sentences. The evaluation assessed the accuracy of these rules individually and revealed that fully automatic conversion of these sentences into a more accessible form is not very reliable.

1 Introduction

People with Autistic Spectrum Disorders (ASD) show a diverse range of reading abilities: on the one hand, 5%-10% of users have the capacity to read words from an early age without the need for formal learning (hyperlexia), on the other hand many users demonstrate weak comprehension of what has been read (Volkmar and Wiesner, 2009). They may have difficulty inferring contextual information or may have trouble understanding mental verbs or emotional language, as well as long sentences with complex syntactic structure (Tager-Flusberg, 1981; Kover et al., 2012). To address these difficulties, the FIRST project[1] is developing a tool which makes texts more accessible for people with ASD. In order to get a better understanding of the needs of these readers, a thorough analysis was carried out to derive a list of high priority obstacles to reading comprehension. Some of these obstacles are related to syntactic complexity and constitute the focus of this paper. Even though the research in the FIRST project focuses on people with ASD, many of the obstacles identified in the project can pose difficulties for a wide range of readers such as language learners and people with other language disorders.

This paper presents and evaluates a set of rules used for simplifying English complex and compound sentences. These rules were developed as part of a syntactic simplification system which was initially developed for users with ASD, but which can also be used for other tasks that require syntactic simplification of sentences. In our research, we consider that syntactic complexity is usually indicated by the occurrence of certain markers or signs of syntactic complexity, referred to hereafter as *signs*, such as punctuation ([,] and [;]), conjunctions ([and], [but], and [or]), complementisers ([that]) or wh-words ([what], [when], [where], [which], [while], [who]). These signs may have a range of syntactic linking and bounding functions which need to be automatically identified, and which we analysed in more detail in (Evans and Orasan, 2013).

Our syntactic simplification process operates in two steps. In the first, signs of syntactic complexity are automatically classified and in the second, manually crafted rules are applied to simplify the relevant sentences. Section 3 presents more details about the method. Evaluation of automatic simplification is a difficult issue. Given that the purpose of this paper is to gain a better understanding of the performance of the rules used for simplifying compound sentences and complex sentences, Section 4 presents the methodology developed for this evaluation and discusses the results obtained. The paper finishes with conclusions.

2 Background information

Despite some findings to the contrary (Arya et al., 2011), automatic syntactic simplification has been motivated by numerous neurolinguistic and psycholinguistic studies. Brain imaging studies indicate that processing syntactically complex struc-

[1] http://first-asd.eu

Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR) @ EACL 2014, pages 131–140,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

tures requires more neurological activity than processing simple structures (Just et al., 1996). A study undertaken by Levy et al. (2012) showed that people with aphasia are better able to understand syntactically simple reversible sentences than syntactically complex ones.

Further motivation is brought by research in NLP, which demonstrates that performance levels in information extraction (Agarwal and Boggess, 1992; Rindflesch et al., 2000; Evans, 2011), syntactic parsing (Tomita, 1985; McDonald and Nivre, 2011), and, to some extent, machine translation (Gerber and Hovy, 1998) are somewhat determined by the length and syntactic complexity of the sentences being processed.

Numerous rule-based methods for syntactic simplification have been developed (Siddharthan, 2006) and used to facilitate NLP tasks such as biomedical information extraction (Agarwal and Boggess, 1992; Rindflesch et al., 2000; Evans, 2011). In these approaches, rules are triggered by pattern-matching applied to the output of text analysis tool such as partial parsers and POS taggers. Chandrasekar and Srinivas (1997) presented an automatic method to learn syntactic simplification rules for use in such systems. Unfortunately, that approach is only capable of learning a restricted range of rules and requires access to expensive annotated resources.

With regard to applications improving text accessibility for human readers, Max (2000) described the use of syntactic simplification for aphasic readers. In work on the PSET project, Canning (2002) implemented a system which exploits a syntactic parser in order to rewrite compound sentences as sequences of simple sentences and to convert passive sentences into active ones for readers with aphasia. The success of these systems is tied to the performance levels of the syntactic parsers that they employ.

More recently, the availability of resources such as *Simple Wikipedia* has enabled text simplification to be included in the paradigm of statistical machine translation (Yatskar et al., 2010; Coster and Kauchak, 2011). In this context, translation models are learned by aligning sentences in *Wikipedia* with their corresponding versions in *Simple Wikipedia*. Manifesting *Basic English* (Ogden, 1932), the extent to which *Simple Wikipedia* is accessible to people with autism has not yet been fully assessed.

The field of text summarisation includes numerous approaches that can be regarded as examples of syntactic simplification. For example, Cohn and Lapata (2009) present a tree-to-tree transduction method that is used to filter non-essential information from syntactically parsed sentences. This compression process often reduces the syntactic complexity of those sentences. An advantage of this approach is that it can identify elements for deletion even when such elements are not indicated by explicit signs of syntactic complexity. The difficulty is that they rely on high levels of accuracy and granularity of automatic syntactic analysis. As noted earlier, it has been observed that the accuracy of parsers is inversely proportional to the length and complexity of the sentences being analysed (Tomita, 1985; McDonald and Nivre, 2011).

The approach to syntactic simplification described in the current paper is a two step process involving detection and tagging of the bounding and linking functions of various signs of syntactic complexity followed by a rule-based sentence rewriting step. Relevant to the first step, Van Delden and Gomez (2002) developed a machine learning method to determine the syntactic roles of commas. Meier et al. (2012) describe German language resources in which the linking functions of commas and semicolons are annotated. The annotated resources exploited by the machine learning method presented in Section 3.2.1 of the current paper are presented in (Evans and Orasan, 2013). From a linguistic perspective, Nunberg et al. (2002) provide a grammatical analysis of punctuation in English.

The work described in this paper was undertaken in a project aiming to improve the accessibility of text for people with autism. It was motivated at least in part by the work of O'Connor and Klein (2004), which describes strategies to facilitate the reading comprehension of people with ASD.

The proposed method is intended to reduce complexity caused by both complex and compound sentences and differs from those described earlier in this section. Sentence compression methods are not suitable for the types of rewriting required in simplifying compound sentences. Parsers are more likely to have lower accuracy when processing these sentences, and therefore the proposed method does not use information about the syntactic structure of sentences in the process. Our method is presented in the next section.

3 The syntactic simplifier

In our research, we regard coordination and subordination as key elements of syntactic complexity. A thorough study of the potential obstacles to the reading comprehension of people with autism highlighted particular types of syntactic complexity, many of which are linked to coordination and subordination. Section 3.1 briefly presents the main obstacles linked to syntactic complexity identified by the study. It should be mentioned that most of the obstacles are problematic not only for autistic people and other types of reader can also benefit from their removal. The obstacles identified constituted the basis for developing the simplification approach briefly described in Section 3.2.

3.1 User requirements

Consultations with 94 subjects meeting the strict DSM-IV criteria for ASD and with IQ $>$ 70 led to the derivation of user preferences and high priority user requirements related to structural processing. A comprehensive explanation of the findings can be found in (Martos et al., 2013). This section discusses briefly the two types of information of relevance to the processing of sentence complexity obtained in our study.

First, in terms of the demand for access to texts of particular genres/domains, it was found that young people (aged 12-16) seek access to documents in informative (arts/leisure) domains and they have less interest in periodicals and newspapers or imaginative texts. Adults (aged 16+) seek access to informative and scientific texts (including newspapers), imaginative text, and the language of social networking and communication. In an attempt to accommodate the interests of both young people and adults, we developed a corpus which contains newspaper articles, texts about health, and literary texts.

Second, the specific morpho-syntactic phenomena that pose obstacles to reading comprehension that are relevant to this paper are:

1. Compound sentences, which should be split into sentences containing a single clause.

2. Complex sentences: in which relative clauses should either be:

   (a) converted into adjectival pre-modifiers or

   (b) deleted from complex sentences and used to generate copular constructions linking the NP in the matrix clause with the predication of the relative clause

In addition, the analysis revealed other types of obstacles such as explicative clauses, which should be deleted, and uncommon conjunctions (including conjuncts) which should be replaced by more common ones. Conditional clauses that follow the main clause and non-initial adverbial clauses should be pre-posed, and passive sentences should be converted in the active form. Various formatting issues such as page breaks that occur within paragraphs and end-of-line hyphenation are also problematic and should be avoided.

Section 3.2 describes the method developed to address the obstacles caused by compound and complex sentences.

3.2 The approach

Processing of obstacles to reading comprehension in this research has focused on detection and reduction of syntactic complexity caused by the occurrence in text of compound sentences (1) and complex sentences (2).

(1) Elaine Trego never bonded with 16-month-old Jacob [and] he was often seen with bruises, a murder trial was told.

(2) The two other patients, who are far more fragile than me, would have been killed by the move.

In (1), the underlined phrases are the conjoins of a coordinate constituent. In (2), the underlined phrase is a subordinate constituent of the larger, superordinate phrase *the two other patients, who are far more fragile than me*.

The overall syntactic simplification pipeline consists of the following steps:

Step 1. Tagging of signs of syntactic complexity with information about their syntactic linking or bounding functions

Step 2. The complexity of sentences tagged in step 1 is assessed and used to trigger the application of two iterative simplification processes, which are applied exhaustively and sequentially to each input sentence:

a. Decomposition of compound sentences (the simplification function converts one input string into two output strings)

b. Decomposition of complex sentences (the simplification function converts one input string into two output strings)

Step 3. Personalised transformation of sentences according to user preference profiles which list obstacles to be tackled and the threshold complexity levels that specify whether simplification is necessary.

Steps 1 and 2 are applied iteratively ensuring that an input sentence can be exhaustively simplified by decomposition of the input string into pairs of progressively simpler sentences. No further simplification is applied to a sentence when the system is unable to detect any signs of syntactic complexity within it. This paper reports on steps 1 and 2. The personalisation step, which takes into consideration the needs of individual users, is not discussed.

3.2.1 Identification of signs of complexity

Signs of syntactic complexity typically indicate constituent boundaries, e.g. punctuation marks, conjunctions, and complementisers. To facilitate information extraction, a rule-based approach to simplify coordinated conjoins was proposed by Evans (2011), which relies on classifying signs based on their linking functions.

In more recent work, an extended annotation scheme was proposed in (Evans and Orasan, 2013) which enables the encoding of links and boundaries between a wider range of syntactic constituents and covers more syntactic phenomena. A corpus covering three text categories (news articles, literature, and patient healthcare information leaflets), was annotated using this extended scheme.[2]

Most sign labels contain three types of information: boundary type, syntactic projection level, and grammatical category of the constituent(s). Some labels cover signs which bound interjections, tag questions, and reported speech and a class denoting false signs of syntactic complexity, such as use of the word *that* as a specifier or anaphor. The class labels are a combination of the following acronyms:

1. $\{C|SS|ES\}$, the generic function as a coordinator (C), the left boundary of a subordinate constituent (SS), or the right boundary of a subordinate constituent (ES).

2. $\{P|L|I|M|E\}$, the syntactic projection level of the constituent(s): prefix (P), lexical (L), intermediate (I), maximal (M), or extended/-clausal (E).

3. $\{A|Adv|N|P|Q|V\}$, the grammatical category of the constituent(s): adjectival (A), adverbial (Adv), nominal (N), prepositional (P), quantificational (Q), and verbal (V).

4. $\{1|2\}$, used to further differentiate subclasses on the basis of some other label-specific criterion.

The scheme uses a total of 42 labels to distinguish between different syntactic functions of the bounded constituents. Although signs are marked by a small set of tokens (words and punctuation), the high number of labels and their skewed distribution make signs highly ambiguous. In addition, each sign is only assigned exactly one label, i.e. that of the dominant constituent in the case of nesting, further increasing ambiguity. These characteristics make automatic classification of signs challenging.

The automatic classification of signs of syntactic complexity is achieved using a machine learning approach described in more detail in Dornescu et al. (2013). After experimenting with several methods of representing the training data and with several classifiers, the best results were obtained by using the BIO model to train a CRF tagger. The features used were the signs' surrounding context (a window of 10 tokens and their POS tags) together with information about the distance to other signs signs in the same sentence and their types. The method achieved an overall accuracy of 82.50% (using 10 fold cross-validation) on the manually annotated corpus.

3.2.2 Rule-based approach to simplification of compound sentences and complex sentences

The simplification method exploits two iterative processes that are applied in sequence to input text that has been tokenised with respect to sentences, words, punctuation, and signs of syntactic complexity. The word tokens in the input text

[2] http://clg.wlv.ac.uk/resources/SignsOfSyntacticComplexity/

Rule ID	CEV-12
Sentence type	Compound (coordination)
Match pattern	A *that* [B] $sign_{CEV}$ [C] .
Transform pattern	A *that* [B]. A *that* [C].
Ex: input	[Investigations showed]$_A$ **that** [the glass came from a car's side window]$_B$ **and**$_{CEV}$ [thousands of batches had been tampered with on five separate weekends]$_C$.
Ex: output	[Investigations showed]$_A$ **that** [the glass came from a car's side window]$_B$. [Investigations showed]$_A$ **that** [thousands of batches had been tampered with on five separate weekends]$_C$.
Rule ID	CEV-26
Sentence type	Compound (coordination)
Match pattern	A v_{CC} B: “[C] $sign_{CEV}$ [D]”.
Transform pattern	A v B: “[C]”. A v B: “[D]”.
Ex: input	[He]$_A$ **added**[]$_B$: “[If I were with Devon and Cornwall police I'd be very interested in the result of this case]$_C$ **and**$_{CEV}$ [I certainly expect them to renew their interest]$_D$.”
Ex: output	[He]$_A$ **added**[]$_B$: “[If I were with Devon and Cornwall police I'd be very interested in the result of this case]$_C$.” [He]$_A$ **added**[]$_B$: “[I certainly expect them to renew their interest]$_D$.”

Table 1: Patterns used to identify conjoined clauses.

have also been labelled with their parts of speech and the signs have been labelled with their grammatical linking and bounding functions. The patterns rely mainly on nine sign labels which delimit clauses (*EV)[3], noun phrases (*MN) and adjectival phrases (*MA). These sign labels can signal either coordinated conjoins (C*) or the start (SS*) or end (ES*) of a constituent.

The first iterative process exploits patterns intended to identify the conjoins of compound sentences. The elements common to these patterns are signs tagged as linking clauses in coordination (label CEV). The second process exploits patterns intended to identify relative clauses in complex sentences. The elements common to these patterns are signs tagged as being left boundaries of subordinate clauses (label SSEV).

The identification of conjoint clauses depends on accurate tagging of words with information about their parts of speech and signs with information about their general roles in indicating the left or right boundaries of subordinate constituents. The identification of subordinate clauses requires more detailed information. In addition to the information required to identify clause conjoins, information about the specific functions of signs is required. The simplification process is thus highly dependent on the performance of the automatic sign tagger.

Table 1 displays two patterns for identifying conjoined clauses and Table 2 displays two patterns for identifying subordinate clauses. In the tables, upper case letters denote contiguous sequences of text,[4] the underbar _ denotes signs of class CEV (in row *Compound*) and SSEV (in row *Complex*). Verbs with clause complements are denoted by v_{CC}, while words of part of speech X are denoted by w_X. The symbol s is used to denote additional signs of syntactic complexity while v denotes words with verbal POS tags. Words explicitly appearing in the input text are italicised. Elements of the patterns representing clause conjoins and subordinate clauses appear in square brackets.

Each pattern is associated with a sentence rewriting rule. A rule is applied on each iteration of the algorithm. Sentences containing signs which correspond to conjoint clauses are converted into two strings which are identical to the original save that, in one, the conjoint clause is replaced by a single conjoin identified in the conjoint while in the other, the identified conjoin is omitted. Sentences containing signs which indicate subordinate clauses are converted into two new strings. One is identical to the original save that the relative clause is deleted. The second is automatically generated, and consists of the NP in the matrix clause modified by the relative clause, a conjugated copula, and the predication of the relative clause. Tables 1 and 2 give examples of transformation rules for the given patterns. In total, 127 different rules were developed for the rewriting of complex sentences and 56 for the rewriting of compound sentences.

[3] In these example the * character is used to indicate any sequence of characters, representing the bounding or linking function of the sign.

[4] Note that these sequences of text may contain additional signs tagged CEV or SSEV.

Rule ID	SSEV-61
Sentence type	Complex (subordination)
Match pattern	A s B [$sign_{SSEV}$ C v D].
Transform pattern	A s B. *That* C v D.
Ex: input	[During the two-week trial, the jury heard how Thomas became a frequent visitor to Roberts's shop in the summer of 1997]$_A$, [after meeting him through a **friend**]$_B$ [**who** [lived near the shop,]$_C$ [described as a "child magnet" by one officer]$_D$.
Ex: output	[During the two-week trial, the jury heard how Thomas became a frequent visitor to Roberts's shop in the summer of 1997]$_A$, [after meeting him through a friend]$_B$. That **friend** [lived near the shop,]$_C$ [described as a "child magnet" by one officer]$_D$.
Rule ID	SSEV-72
Sentence type	Complex (subordination)
Match pattern	[A w_{IN} w_{DT}* n $\{n \mid of\}$* $sign_{SSEV}$] w_{VBD} B $\{.\mid?\mid!\}$
Transform pattern	N/A
	Pattern SSEV-72 is used to prevent rewriting of complex sentences when the subordinate clause is the argument of a clause complement verb. The result of this rule is to strip the tag from the triggering sign of syntactic complexity
Ex: input	[Eamon Reidy, 32,]$_A$ fled [across fields in Windsor Great Park after the crash[**,** the court heard.]

Table 2: Patterns used to identify subordinate clauses.

4 Evaluation

The detection and classification of signs of syntactic complexity can be evaluated via standard methods in LT based on comparing classifications made by the system with classifications made by linguistic experts. This evaluation is reported in (Dornescu et al., 2013). Unfortunately, the evaluation of the actual simplification process is difficult, as there are no well established methods for measuring its accuracy. Potential methodologies for evaluation include comparison of system output with human simplification of a given text, analysis of the post-editing effort required to convert an automatically simplified text into a suitable form for end users, comparisons using experimental methods such as eye tracking and extrinsic evaluation via NLP applications such as information extraction, all of which have weaknesses in terms of adequacy and expense.

Due to the challenges posed by these previously established methods, we decided that before we employ them and evaluate the output of the system as a whole, we focus first on the evaluation of the accuracy of the two rule sets employed by the syntactic processor. The evaluation method is based on comparing sets of simplified sentences derived from an original sentence by linguistic experts with sets derived by the method described in Section 3.

4.1 The gold standard

Two gold standards were developed to support evaluation of the two rule sets. Texts from the genres of health, literature, and news were processed by different versions of the syntactic simplifier. In one case, the only rules activated in the syntactic simplifier were those concerned with rewriting compound sentences. In the second case, the only rules activated were those concerned with rewriting complex sentences. The output of the two versions was corrected by a linguistic expert to ensure that each generated sentence was grammatically well-formed and consistent in meaning with the original sentence. Sentences for which even manual rewriting led to the generation of grammatically well-formed sentences that were not consistent in meaning with the originals were removed from the test data. After filtering, the test data contained nearly 1,500 sentences for use in evaluating rules to simplify of compound sentences, and nearly 1,100 sentences in the set used in evaluating rules to simplify complex sentences. The break down per genre/domain is given in Tables 3a and 3b.

The subset of sentences included in the gold standard contained manually annotated information about the signs of syntactic complexity. This was done to enable reporting of the evaluation results in two modes: one in which the system consults an oracle for classification of signs of syntactic complexity and one in which the system consults the output of the automatic sign tagger.

4.2 Evaluation results

Evaluation results are reported in terms of accuracy of the simplification process and the change in readability of the generated sentences. Computation of accuracy is based on the mean Leven-

		Text category		
		News	Health	Literature
#Compound sentences		698	325	418
Accuracy	Oracle	0.758	0.612	0.246
	Classifier	0.314	0.443	0.115
ΔFlesch	Oracle	11.1	8.2	15.3
	Classifier	9.9	10.2	13.6
ΔAvg.	Oracle	-12.58	-9.86	-16.69
Sent. Len.	Classifier	-13.08	-12.30	-16.79

(a) Evaluation of simplification of compound sentences

		Text category		
		News	Health	Literature
#Complex sentences		369	335	379
Accuracy	Oracle	0.452	0.292	0.475
	Classifier	0.433	0.227	0.259
ΔFlesch	Oracle	2.5	0.8	2.3
	Classifier	2.3	0.9	2.3
ΔAvg.	Oracle	-2.96	-0.90	-2.80
Sent. Len.	Classifier	-2.80	-0.99	-2.11

(b) Evaluation of simplification of complex sentences

Table 3: Evaluation results for the two syntactic phenomena on three text genres

shtein similarity[5] between the sentences generated by the system and the most similar simplified sentences verified by the linguistic expert. Once the most similar sentence in the key has been found, that element is no longer considered for the rest of the simplified sentences in the system's response to the original. In this evaluation, sentences are considered to be converted correctly if their LS $>$ 0.95. The reason for setting such a high threshold for the Levenshtein ratio is because the evaluation method should only reward system responses that match the gold standard almost perfectly save for a few characters which could be caused by typos or variations in the use of punctuation and spaces. A sentence is considered successfully simplified, and implicitly all the rules used in the process are considered correctly applied, when all the sentences produced by the system are converted correctly according to the gold standard. This evaluation approach may be considered too inflexible as it does not take into consideration the fact that a sentence can be simplified in several ways. However, the purpose here is to evaluate the way in which sentences are simplified using specific rules.

In order to calculate the readability of the generated sentences we initially used the Flesch score (Flesch, 1949). However, our system changes the text only by rewriting sentences into sequences of simpler sentences and does not make any changes at the lexical level. For this reason, any changes observed in the Flesch score are due to changes in the average sentence length. Therefore, for our experiments we report both ΔFlesch score and Δaverage sentence length.

The evaluation results are reported separately for the three domains. In addition, the results are calculated when the classes of the signs are derived from the manually annotated data (*Oracle*) and from use of the automatic classifier (*Classifier*).

Table 3a presents the accuracy of the rules implemented to convert compound sentences into a more accessible form. The row *#Compound sentences* displays the number of sentences in the test data that contain signs of conjoint clauses (signs of class CEV). The results obtained are not unexpected. In all cases the accuracy of the simplification rules is higher when the labels of signs are assigned by the oracle. With the exception of the health domain, the same pattern is observed when ΔFlesch is considered. The highest accuracy is obtained on the news texts, then the health domain, and finally the literature domain. However, despite significantly lower accuracy on the literature domain, the readability of the sentences from the literature domain benefits most from the automatic simplification. This can be noticed both in the improved Flesch scores and reduced sentence length.

Table 3b presents the accuracy of the rules which simplify complex sentences. In this table, *#Complex sentences* denotes the number of sentences in the test data that contain relative clauses. The rest of the measures are calculated in the same way as in Table 3a. Inspection of the table shows that, for the news and health domains, the accuracy of these simplification rules is significantly lower than the simplification rules used for compound sentences. Surprisingly, the rules work better for the literature domain than for the others. The improvement in the readability of texts from the health domain is negligible, which can be explained by the poor performance of the simplification rules on this domain.

[5] Defined as 1 minus the ratio of Levenshtein distance between the two sentences to the length in characters of the longest of the two sentences being compared.

4.3 Error analysis

In order to have a better understanding of the performance of the system, the performance of the individual rules was also recorded. Tables 4 and 5 contain the most error prone trigger patterns for conjoined and subordinate clauses respectively. The statistics were derived from rules applied to texts of all three categories of texts and the signs of syntactic complexity were classified using an oracle, in order to isolate the influence of the rules in the system output. In this context, the accuracy with which the syntactic processor converts sentences containing conjoint clauses into a more accessible form is 0.577. The accuracy of this task with regard to subordinate clauses is 0.411.

The most error-prone trigger patterns for conjoined clauses are listed in Table 4, together with information on the conjoin that they are intended to detect (left or right), their error rate, and the number of number of errors made. The same information is presented for the rules converting sentences containing subordinate clauses in Table 5, but in this case the patterns capture the subordination relations. In the patterns, words with particular parts of speech are denoted by the symbol w with the relevant Penn Treebank tag appended as a subscript. Verbs with clause complements are denoted v_{CC}. Signs of syntactic complexity are denoted by the symbol *s* with the abbreviation of the functional class appended as a subscript. Specific words are printed in italics. In the patterns, the clause coordinator is denoted '_' and upper case letters are used to denote stretches of contiguous text.

Rules CEV-25a and SSEV-78a are applied when the input sentence triggers none of the other implemented patterns. Errors of this type quantify the number of sentences containing conjoint or subordinate clauses that cannot be converted into a more accessible form by rules included in the structural complexity processor. Both rules have quite high error rates, but these errors can only be addressed via the addition of new rules or the adjustment of already implemented rules.

SSEV-36a is a pattern used to prevent processing of sentences that contain verbs with clause complements. This pattern was introduced because using the sentence rewriting algorithm proposed here to process sentences containing these subordinate clauses would generate ungrammatical output.

Table 5 contains only 4 items because for the rest of the patterns the number of errors was less than 3. A large number of these rules had an error rate of 1 which motivated their deactivation. Unfortunately this did not lead to improved accuracy of the overall conversion process.

5 Conclusions and future work

Error analysis revealed that fully automatic conversion compound and complex sentences into a more accessible form is quite unreliable, particularly for texts of the literature category. It was noted that conversion of complex sentences into a more accessible form is more difficult than conversion of compound sentences. However, subordinate clauses are significantly more prevalent than conjoint clauses in the training and testing data collected so far.

The evaluation of the rule sets used in the conversion of compound and complex sentences into a more accessible form motivates further specific development of the rule sets. This process includes deletion of rules that do not meet particular thresholds for accuracy and the development of new rules to address cases where input sentences fail to trigger any conversion rules (signalled by activation of redundant rules CEV-25a and SSEV-78a).

The results are disappointing given that the syntactic simplification module presented in this paper is expected to be integrated in a system that makes texts more accessible for people with autism. However, this simplification module will be included in a post-editing environment for people with ASD. In this setting, it may still prove useful, despite its low accuracy.

Acknowledgments

The research described in this paper was partially funded by the European Commission under the Seventh (FP7-2007-2013) Framework Programme for Research and Technological Development (FP7- ICT-2011.5.5 FIRST 287607). We gratefully acknowledge the contributions of all the members of the FIRST consortium for their feedback and comments during the development of the methods, and to Laura Hasler for her help with the evaluation.

ID	Conjoin	Trigger pattern	Error rate	#Errors
CEV-24b	B	A _ B	0.131	59
CEV-24a	A	A _ B	0.119	54
CEV-12b	A that C	A *that* B _ C	0.595	25
CEV-25a	NA	NA	0.956	22
CEV-26a	A v_{CCV} B : “C”	A v_{CC} B : “C _ D”	0.213	16
CEV-26b	A v_{CCV} B : “D”	A v_{CC} B : “C _ D”	0.203	14

Table 4: Error rates for rules converting sentences with conjoint clauses

ID	Matrix clause / subordinate clause	Trigger pattern	Error rate	#Errors
SSEV-78a	NA	NA	0.517	45
SSEV-72a	A , _ C $w_{\{verb\}}$ D	A s B _ C $w_{\{verb\}}$ D	0.333	4
SSEV-36a	NA	A told $w_{\{noun \mid PRP \mid DT \mid IN\}}$ * _ B	0.117	4
SSEV-13b	$w_{VBN}\ w_{IN}\ (w_{\{DT \mid PRP\$ \mid noun \mid CD\}} \mid\text{-}\mid,)^*\ w_{\{noun\}}$ B	A $w_{VBN}\ w_{IN}\ \{w_{\{DT \mid PRP\$ \mid noun \mid CD\}} \mid\text{-}\mid,\}^*\ w_{\{noun\}}$ _ B	1	3

Table 5: Error rates for rules converting sentences with subordinate clauses

References

Rajeev Agarwal and Lois Boggess. 1992. A simple but useful approach to conjunct identification. In *Proceedings of the 30th annual meeting for Computational Linguistics*, pages 15–21, Newark, Delaware. Association for Computational Linguistics.

D. J. Arya, Elfrieda H. Hiebert, and P. D. Pearson. 2011. The effects of syntactic and lexical complexity on the comprehension of elementary science texts. *International Electronic Journal of Elementary Education*, 4 (1):107–125.

Y. Canning. 2002. *Syntactic Simplification of Text*. Ph.d. thesis, University of Sunderland.

R Chandrasekar and B Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10:183–190.

T. Cohn and M. Lapata. 2009. Sentence Compression as Tree Transduction. *Journal of Artificial Intelligence Research*, 20(34):637–74.

W. Coster and D. Kauchak. 2011. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, pages 665–669, Portland, Oregon, June. Association of Computational Linguistics.

Iustin Dornescu, Richard Evans, and Constantin Orăsan. 2013. A Tagging Approach to Identify Complex Constituents for Text Simplification. In *Proceedings of Recent Advances in Natural Language Processing*, pages 221 – 229, Hissar, Bulgaria.

Richard Evans and Constantin Orasan. 2013. Annotating signs of syntactic complexity to support sentence simplification. In I. Habernal and V. Matousek, editors, *Text, Speech and Dialogue. Proceedings of the 16th International Conference TSD 2013*, pages 92–104. Springer, Plzen, Czech Republic.

R. Evans. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*, 26 (4):371–388.

R. Flesch. 1949. *The art of readable writing*. Harper, New York.

Laurie Gerber and Eduard H. Hovy. 1998. Improving translation quality by manipulating sentence length. In David Farwell, Laurie Gerber, and Eduard H. Hovy, editors, *AMTA*, volume 1529 of *Lecture Notes in Computer Science*, pages 448–460. Springer.

M. A. Just, P. A. Carpenter, and K. R. Thulborn. 1996. Brain activation modulated by sentence comprehension. *Science*, 274:114–116.

S. T. Kover, E. Haebig, A. Oakes, A. McDuffie, R. J. Hagerman, and L. Abbeduto. 2012. Syntactic comprehension in boys with autism spectrum disorders: Evidence from specific constructions. In *Proceedings of the 2012 International Meeting for Autism Research*, Athens, Greece. International Society for Autism Research.

J. Levy, E. Hoover, G. Waters, S. Kiran, D. Caplan, A. Berardino, and C. Sandberg. 2012. Effects of syntactic complexity, semantic reversibility, and explicitness on discourse comprehension in persons with aphasia and in healthy controls. *American Journal of Speech–Language Pathology*, 21(2):154 – 165.

Wolfgang Maier, Sandra Kübler, Erhard Hinrichs, and Julia Kriwanek. 2012. Annotating coordination in the penn treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 166–174, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Juan Martos, Sandra Freire, Ana Gonzlez, David Gil, Richard Evans, Vesna Jordanova, Arlinda Cerga, Antoneta Shishkova, and Constantin Orasan. 2013. User preferences: Updated report. Technical report, The FIRST Consortium, Available at http://first-asd.eu/D2.2.

A. Max. 2000. *Syntactic simplification - an application to text for aphasic readers*. Mphil in computer speech and language processing, University of Cambridge, Wolfson College.

Ryan T. McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Geoffrey Nunberg, Ted Briscoe, and Rodney Huddleston. 2002. Punctuation. chapter 20 In Huddleston, Rodney and Geoffrey K. Pullum (eds) *The Cambridge Grammar of the English Language*, pages 1724–1764. Cambridge University Press.

I. M. O'Connor and P. D. Klein. 2004. Exploration of strategies for facilitating the reading comprehension of high-functioning students with autism spectrum disorders. *Journal of Autism and Developmental Disorders*, 34:2:115–127.

C. K. Ogden. 1932. *Basic English: a general introduction with rules and grammar*. K. Paul, Trench, Trubner & Co., Ltd., London.

Thomas C. Rindflesch, Jayant V. Rajan, and Lawrence Hunter. 2000. Extracting molecular binding relationships from biomedical text. In *Proceedings of the sixth conference on Applied natural language processing*, pages 188–195, Seattle, Washington. Association of Computational Linguistics.

A. Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4:1:77–109.

Helen Tager-Flusberg. 1981. Sentence comprehension in autistic children. *Applied Psycholinguistics*, 2:1:5–24.

Masaru Tomita. 1985. *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems*. Kluwer Academic Publishers, Norwell, MA, USA.

Sebastian van Delden and Fernando Gomez. 2002. Combining finite state automata and a greedy learning algorithm to determine the syntactic roles of commas. In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '02, pages 293–, Washington, DC, USA. IEEE Computer Society.

F.R. Volkmar and L. Wiesner. 2009. *A Practical Guide to Autism*. Wiley, Hoboken, NJ.

M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 365–368, Los Angeles, California, June. Association of Computational Linguistics.