

Text simplification using synchronous dependency grammars: Generalising automatically harvested rules

M.A. Angrosh

Computing Science
University of Aberdeen, UK
angroshmandya@abdn.ac.uk

Advaith Siddharthan

Computing Science
University of Aberdeen, UK
advait@abdn.ac.uk

Abstract

We present an approach to text simplification based on synchronous dependency grammars. Our main contributions in this work are (a) a study of how automatically derived lexical simplification rules can be generalised to enable their application in new contexts without introducing errors, and (b) an evaluation of our hybrid system that combines a large set of automatically acquired rules with a small set of hand-crafted rules for common syntactic simplification. Our evaluation shows significant improvements over the state of the art, with scores comparable to human simplifications.

1 Introduction

Text simplification is the process of reducing the linguistic complexity of a text, while still retaining the original information content and meaning. Text Simplification is often thought of as consisting of two components - syntactic simplification and lexical simplification. While syntactic simplification aims at reducing the grammatical complexity of a sentence, lexical simplification focuses on replacing difficult words or short phrases by simpler variants.

Traditionally, entirely different approaches have been used for lexical (Devlin and Tait, 1998; Biran et al., 2011; Yatskar et al., 2010; Specia et al., 2012) and syntactic simplification (Canning, 2002; Chandrasekar et al., 1996; Siddharthan, 2011; De Belder and Moens, 2010; Candido Jr et al., 2009). Recent years have seen the application of machine translation inspired approaches to text simplification. These approaches learn from aligned English and Simplified English sentences extracted from the Simple English Wikipedia (SEW) corpus (simple.wikipedia.org). However, even these approaches (Woodsend and Lapata,

2011; Wubben et al., 2012; Coster and Kauchak, 2011; Zhu et al., 2010) struggle to elegantly model the range of lexical and syntactic simplification operations observed in the monolingual simplification task within one framework, often differentiating between operation at leaf nodes of parse trees (lexical) and internal tree nodes (syntactic). The key issue is the modelling of context for application of lexical rules. While syntactic rules (for splitting conjoined clauses, or disembedding relative clauses) are typically not context dependent, words are typically polysemous and can only be replaced by others in appropriate contexts.

Our main contribution in this paper is to present a unified framework for representing rules for syntactic and lexical simplification (including paraphrase involving multiple words), and study for the first time how the definition of context affects system performance. A second contribution is to provide a substantial human evaluation (63 sentences and 70 participants) to evaluate contemporary text simplification systems against manually simplified output.

2 Related work

Text simplification systems are characterised by the level of linguistic knowledge they encode, and by whether their simplification rules are hand-crafted or automatically acquired from a corpus.

In recent times, the availability of a corpus of aligned English Wikipedia (EW) and Simple English Wikipedia (SEW) sentences has led to the application of various “monolingual translation” approaches to text simplification. Phrase Based Machine Translation (PBMT) systems (Specia, 2010; Coster and Kauchak, 2011; Wubben et al., 2012) use the least linguistic knowledge (only word sequences), and as such are ill equipped to handle simplifications that require morphological changes, syntactic reordering or sentence splitting.

Zhu et al. (2010) in contrast present an approach based on syntax-based SMT (Yamada and

Knight, 2001). Their translation model encodes probabilities for four specific rewrite operations on the parse trees of the input sentences: substitution, reordering, splitting, and deletion. Woodsend and Lapata (2011) propose quasi-synchronous tree substitution grammars (QTSG) for a similarly wide range of simplification operations as well as lexical substitution. Narayan and Gardent (2014) combine PMBT for local paraphrase with a syntactic splitting component based on a deep semantic representation. None of these systems model morphological information, which means some simplification operations such as voice conversion cannot be handled correctly.

Against this limitation, hand-crafted systems have an advantage here, as they tend to encode the maximum linguistic information. We have previously described systems (Siddharthan, 2010; Siddharthan, 2011) that can perform voice conversion accurately and use transformation rules that encode morphological changes as well as deletions, re-orderings, substitutions and sentence splitting. On the other hand, such hand-crafted systems are limited in scope to syntactic simplification as there are too many lexico-syntactic and lexical simplifications to enumerate by hand. We have also previously described how to construct a hybrid system that combines automatically derived lexical rules with hand-crafted syntactic rules within a single framework (Siddharthan and Mandya, 2014). We extend that work here by describing how such automatically learnt rules can be generalised.

3 Simplification using synchronous dependency grammars

We follow the architecture proposed in Ding and Palmer (2005) for Synchronous Dependency Insertion Grammars, reproduced in Fig. 1.

In this paper, we focus on the decomposition of a dependency parse into Elementary Trees (ETs), and the learning of rules to transduce a source ET to a target ET. We use the datasets of Coster and Kauchak (2011) and Woodsend and Lapata

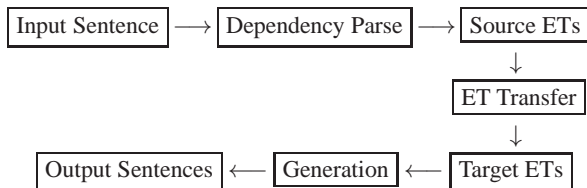


Figure 1: System Architecture

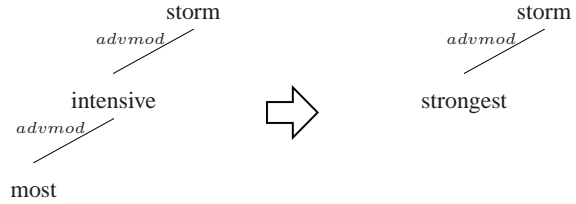


Figure 2: Transduction of Elementary Trees (ETs)

(2011) for learning rules. These datasets consist of $\sim 140K$ aligned simplified and original sentence pairs obtained from Simple English Wikipedia and English Wikipedia. The rules are acquired in the format required by the RegenT text simplification system (Siddharthan, 2011), which is used to implement the simplification. This requires dependency parses from the Stanford Parser (De Marneffe et al., 2006), and generates output sentences from dependency parses using the generation-light approach described in (Siddharthan, 2011).

3.1 Acquiring rules from aligned sentences

To acquire a synchronous grammar from dependency parses of aligned English and simple English sentences, we just need to identify the differences. For example, consider two aligned sentences from the aligned corpus described in Woodsend and Lapata (2011):

1. (a) It was the second most intensive storm on the planet in 1989.
- (b) It was the second strongest storm on the planet in 1989.

An automatic comparison of the dependency parses for the two sentences (using the Stanford Parser) reveals that there are two typed dependencies that occur only in the parse of the first sentence, and one that occur only in the parse of the second sentence. Thus, to convert the first sentence into the second, we need to delete two dependencies and introduce one other. From this example, we extract the following rule:

RULE 1: MOST_INTENSIVE2STRONGEST

1. DELETE
 - (a) $\text{advmod}(\text{?X0}[\text{intensive}], \text{?X1}[\text{most}])$
 - (b) $\text{advmod}(\text{?X2}[\text{storm}], \text{?X0}[\text{intensive}])$
2. INSERT
 - (a) $\text{advmod}(\text{?X2}, \text{?X3}[\text{strongest}])$

The rule contains variables (?X_n), which can be forced to match certain words in square brackets.

Such deletion and insertion operations are central to text simplification, but a few other operations are also needed to avoid broken dependency links in the Target ETs. These are enumerated in (Siddharthan, 2011) and will not be reproduced here for shortage of space. By collecting such rules, we can produce a meta-grammar that can translate dependency parses in one language (English) into the other (simplified English). The rule above will translate “most intensive” to “strongest”, in the immediate lexical context of “storm”. For ease of presentation, we present the ET Transfer component as transformation rules, but this rule can also be presented as a transduction of elementary trees (Fig. 2).

3.2 Generalising rules

It is clear that the rule shown above will only be applied if three different words (“storm”, “most” and “intensive”) occur in the exact syntax specified on the left hand side of Figure 2. The rule is correct, but of limited use, for “most intensive” can be simplified to “strongest” only when it modifies the word “storm”.

The modelling of lexical context is a particular weak point in previous work; for instance, Woodsend and Lapata (2011), in their quasi-synchronous tree substitution grammar, remove all lexical context for lexical simplification rules, to facilitate their application in new contexts. Similarly, phrase-based machine translation can default to lexical simplification using word level alignments if longer substrings from the input text are not found in the alignment table. However, as words can have different senses, lexical substitution without a lexical context model is error prone.

Our goals here are to enumerate methods to generalise rules, and to evaluate performance on unseen sentences. All the methods described are automated, and do not require manual effort.

Generalising from multiple instances: A single rule can be created from multiple instances in the training data. For example, if the modifier “extensive” has been simplified to “big” in the context of a variety of words in the ?X0 position, this can be represented succinctly as “?X0[networks, avalanches, blizzard, controversy]”. Note that this list provides valid lexical contexts for application of the rule. If the word is seen in sufficient contexts, we make it universal by removing the list. Rule 2 below states that any of the words in “[ex-

tensive, large, massive, sizable, major, powerful, giant]” can be replaced by “big” in any lexical context ?X0, provided the syntactic context is an *amod* relation. To de-lexicalise context in this manner, each lexical substitution needs to have been observed in 10 different contexts. While not foolproof, this ensures that lexical context is removed only for common simplifications, which are more likely to be independent of context.

RULE 2: *2BIG

1. DELETE
 - (a) *amod*(?X0, ?X1[extensive, large, massive, sizable, major, powerful, giant])
2. INSERT
 - (a) *amod*(?X0, ?X2[big])

Reducing context size: Often, single lexical changes result in multiple relations in the INSERT and DELETE lists. Rule 3 shows a rule where the verb “amend” has been simplified to “change”, in a context where the direct object is “Constitution” and there is an infinitive modifier relation to “proposals”, using the auxiliary “to”.

RULE 3: AMEND2CHANGE

1. DELETE
 - (a) *aux*(?X0[amend], ?X1[to])
 - (b) *infmod*(?X2[proposals], ?X0[amend])
 - (c) *doj*(?X0[amend], ?X3[Constitution])
2. INSERT
 - (a) *aux*(?X4[change], ?X1)
 - (b) *infmod*(?X2, ?X4)
 - (c) *doj*(?X4, ?X3)
3. MOVE
 - (a) ?X0 ?X4

Rule 3 also shows the MOVE command created to move any other relations (edges) involving the node ?X0 to the newly created node ?X4. The MOVE list is automatically created when a variable (?X0) is present in the DELETE list but not in the INSERT list and ensures correct rule application in new contexts where there might be additional modifiers connected to the deleted word.

Rule 3 clearly encodes too much context. In such cases, we reduce the context by creating three rules, each with a reduced context of one relation (*aux*, *infmod* or *doj*); for example:

RULE 3A: AMEND2CHANGE3

1. DELETE: *doj*(?X0[amend], ?X1[Constitution])
2. INSERT: *doj*(?X2[change], ?X1)
3. MOVE: ?X0 ?X2

In this paper, we generate rules with each possible lexical context, but one could filter out relations such as *aux* that provide a lexical context of a closed class word. The generalised Rule 3A makes clear the need for the MOVE operation, which is implemented in RegenT by rewriting ?X0 as ?X2 in the entire dependency parse after rule application. We will omit the MOVE command where it is not required to save space.

Extracting elementary trees: It is possible for the DELETE and INSERT lists to contain multiple simplification rules; i.e., multiple transductions over ETs (connected graphs). We need to ensure that each extracted rule contains a connected graph in the DELETE list. Where this is not the case, we split the rule into multiple rules. An example follows where three independent simplifications have been performed on a sentence:

4. (a) As a general rule , with an increase in elevation comes a decrease in temperature and an increase in precipitation .
- (b) As a normal rule , with an increase in height comes a decrease in temperature and an increase in rain .

The original extracted rule contains three relations with no variable in common:

RULE 4: INDEPENDENTELEMENTARYTREES

1. DELETE
 - (a) amod(?X0[rule], ?X1[general])
 - (b) prep_in(?X2[comes], ?X3[elevation])
 - (c) prep_in(?X4[increase], ?X5[precipitation])
2. INSERT
 - (a) amod(?X0, ?X6[normal])
 - (b) prep_in(?X2, ?X7[height])
 - (c) prep_in(?X4, ?X8[rain])

Relations with no variables in common belong to separate ETs, so we create three new rules:

- RULE 4A
 DELETE: amod(?X0[rule], ?X1[general])
 INSERT: amod(?X0, ?X6[normal])
- RULE 4B
 DELETE: prep_in(?X2[comes], ?X3[elevation])
 INSERT: prep_in(?X2, ?X7[height])
- RULE 4C
 DELETE: prep_in(?X4[increase], ?X5[precipitation])
 INSERT: prep_in(?X4, ?X8[rain])

Removing lexical context from longer rules: While preserving lexical context is important to avoid meaning change in new contexts due to polysemy (this claim is evaluated in §3.5), it is unnecessary for longer rules involving more than one relation, as these tend to encode longer paraphrases with more standardised meanings. We thus remove the lexical context for rules involving multiple relations in the DELETE list¹.

3.3 Overview of extracted ruleset

In addition to the generalisation steps described above, we also automatically filtered out rules that were undesired for various reasons. As we use manually written rules in RegenT for common syntactic simplification (as described in Sidharthan (2011)), we filter out rules that involve dependency relations for passive voice, relative clauses, apposition, coordination and subordination. We also filter out rules with relations that are error-prone, based on a manual inspection. These involved single lexical changes involving the following dependencies: *det* and *num* (rules that change one determiner to another, or one number to another) and *poss* and *pobj* that mostly appeared in rules due to errorful parses. We also automatically filtered out errorful rules using the training set as follows: we applied the rules to the source sentence from which they were derived, and filtered out rules that did not generate the target sentence accurately. Finally, we restricted the number of relations in either the DELETE or INSERT list to a maximum of three, as longer rules were never being applied.

Tab. 1 shows how the filters and generalisation influence the number of rules derived involving 1–5 relations in each of the DELETE and INSERT lists. In addition, we also extract rules where the DELETE list is longer than the INSERT list; i.e., simplification that result in sentence shortening (e.g., Rule 1 in Section 3.1).

Tab. 2 provides details of the final number of filtered and generalised rules for different lengths of the DELETE and INSERT lists. The ruleset shown in Tab. 2 will henceforth be referred to as WIKI.

3.4 Generalising context with WordNet

To generalise the context of lexical simplification rules further, we now consider the use of WordNet

¹Lexical context is defined as lexical specifications on variables occurring in both the DELETE and INSERT lists; i.e., words that are unchanged by the simplification.

DELETE	INSERT	IS	FS	GS
1	1	1111	593	4250
2	2	1051	357	171
3	3	1108	178	52
4	4	831	-	-
5	5	628	-	-

Table 1: Number of extracted rules where the INSERT and DELETE lists contain 1–5 relations (IS: initial set; FS: filtered set; GS: generalised set)

in expanding lexical context. The idea is that the lexical specification of context variables in rules can be expanded by identifying related words in WordNet. We propose to use Lin’s similarity measure (Lin, 1998), an information content based similarity measure for our experiments as information content based measures are observed to perform better in deriving similar terms, in comparison to other methods (Budanitsky and Hirst, 2006). Lin’s formula is based on Resnik’s. Let $IC(c) = -\log p(c)$ be the information content of a concept (synset) in WordNet, where $p(c)$ is the likelihood of seeing the concept (or any of its hyponyms) in a corpus. Resnik defines the similarity of two concepts c_1 and c_2 as $sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} IC(c)$, the IC of the most specific class c that subsumes both c_1 and c_2 . Lin’s formula normalises this by the IC of each class:

$$sim_{lin}(c_1, c_2) = \frac{2 \cdot sim_{res}(c_1, c_2)}{IC(c_1) + IC(c_2)}$$

Our next goal is to explore how the definition of lexical context impacts on a text simplification system.

3.5 Evaluation

To evaluate our work, we used the text simplification tool RegenT (Siddharthan, 2011) to apply different versions of the acquired rule sets to a test dataset. For example, consider the following rule shown in 6(a). This is the original rule extracted from the training data (cf. Tab. 2).

RULE 6(A): RULE-WIKI

1. DELETE
 - (a) nsubjpass(??X0[adapted], ??X1[limbs])
2. INSERT
 - (a) nsubjpass(??X0, ??X2[legs])

This rule is transformed to a no-context rule in 6(b), where words such as “adapted” that occur in

DELETE / INSERT	1	2	3	4	5
1 Relation	4250				
2 Relations	110	171			
3 Relations	91	165	52		
4 Relations	49	71	209	-	
5 Relations	24	44	80	-	-

Table 2: Details of rules derived with different length in DELETE and INSERT relations

both the DELETE and INSERT lists are removed entirely from the rule:

RULE 6(B): NO-CONTEXT

1. DELETE
 - (a) nsubjpass(??X0, ??X1[limbs])
2. INSERT
 - (a) nsubjpass(??X0, ??X2[legs])

Finally the rule in 6(c), expands the context word “adapted” using WordNet classes with Lin’s similarity greater than 0.1.

RULE 6(C): RULE-WITH-WORDNET0.1-CONTEXT

1. DELETE
 - (a) nsubjpass(??X0[accommodated,adapted,adjusted,altered,assimilated,changed,complied,conformed,fited,followed,gearred,heeded,listened,minded,moved,obeyed,oriented,pitched,tailored,varied], ??X1[limbs])
2. INSERT
 - (a) nsubjpass(??X0, ??X2[legs])

Evaluation of generalisability of rules: We expanded the context of rules derived from Wikipedia using various thresholds such as 0.1, 0.4 and 0.8 for Lin similarity measure and evaluated how many simplification operations were performed on the first 11,000 sentences from the dataset of Coster and Kauchak (2011). The details of rules applied on the test dataset, using different thresholds along with the Wiki-context and no-context rules are provided in Tab. 3. As seen, there is an increase in the application of rules with the decrease in threshold for Lin similarity measure. Removing the lexical context entirely results in an even larger increase in rule application. Next, we evaluate the correctness of rule application.

Evaluation of correctness of rule application: To test the correctness of the rule applications with

Rule Version	Rules	% Change
Wikicontext	7610	
WordNet context (0.8)	7870	3.41
WordNetcontext (0.4)	8488	11.85
WordNetcontext (0.1)	10715	40.80
Nocontext	31589	315.09

Table 3: Application of different versions of rules on test dataset (% change is the increase in the application of rules between Wiki-context and the corresponding version)

different rule sets, we performed a human evaluation to gauge how fluent and simple the simplified sentences were, and to what extent they preserved the meaning of the original. We compared three versions in this experiment: the original ruleset, the context expanded using $Sim_{Lin} \geq 0.1$ (40% increase in rule applications) and with no lexicalised context (315% increase in rule applications). The goal is to identify a level of generalisation that increases rule application in new contexts without introducing more errors.

We used the first 11,000 sentences from the dataset of Coster and Kauchak (2011), the same dataset used for rule acquisition. We extracted at random 30 sentences where a simplification had been performed using the original ruleset. This gives an upper bound on the performance of the original Wikipedia-context ruleset, as these are all sentences from which the rules have been derived.

We then selected a further 30 sentences where a simplification had been performed using the WordNet-context ($Lin=0.1$), but not with the original ruleset. These are new applications of the generalised ruleset on sentences that it hasn't directly learnt rules from. Similarly, we selected a further 30 sentences where a simplification had been performed using the no-context ruleset, but not the Wikipedia-context or WordNet-context rulesets. Thus each set of 30 sentences contains new applications of the ruleset, as the lexical context is expanded, or abandoned completely.

This process gave us a total of 90 sentences to evaluate. We recruited participants through Amazon Mechanical Turk. Participants were filtered to be in the US and have an approval rating of 80%. These raters were shown 30 examples, each containing an original Wikipedia sentence followed by one of the simplified versions (WI, WN or NC). Order of presentation was random. For each such pair, raters were asked to rate each simplified version for fluency, simplicity and the extent to which

it preserved the meaning of the original. The experiment provided 917 ratings for 90 sentences involving 28 raters. We used a Likert scale of 1–5, where 1 is totally unusable output, and 5 is the output that is perfectly usable.

The mean values and the standard deviation for fluency, simplicity and meaning preservation for sentences simplified using WordNet ($Lin=0.1$), Wiki and no context is shown in Tab. 4. As seen, the difference between the mean values for all three criteria of fluency, simplicity and meaning preservation between WordNet and Wiki version is very small as compared to simplified sentences with no-context rules. An analysis of variance (ANOVA) test was conducted to measure the effect of fluency, simplicity and meaning preservation for versions of simplified text.

Fluency: A one-way ANOVA conducted to evaluate fluency for versions of simplified text showed a highly significant effect of version (WN, WC, and NC) on the fluency score ($F=51.54$, $p=2 \times 10^{-16}$). A Tukey's pairwise comparison test (Tukey's HSD, overall alpha level = 0.05) indicated significant difference between WI and NC and between WN and NC at $p = 0.01$. However, the difference between WN and WI was not significant at $p = 0.01$.

Simplicity: The ANOVA conducted to evaluate simplicity for different versions also showed a significant effect of version on the simplicity score ($F=76.7$, $p=2 \times 10^{-16}$). A Tukey's pairwise comparison test (Tukey's HSD, overall alpha level = 0.05) indicated significant difference between WN and NC and WI and NC ($p < 0.01$). However, the difference between WN and WI was not significant at $p = 0.01$.

Meaning Preservation: The ANOVA conducted to evaluate meaning preservation for versions of simplified text also showed a highly significant effect of version on the meaning preservation score ($F=17.22$, $p=4.55 \times 10^{-8}$). A Tukey's pairwise comparison test (Tukey's HSD, overall alpha level = 0.05) indicated significant difference between WN and NC and WI and NC ($p < 0.01$). However, the difference between WN and WI was not significant at $p = 0.01$.

This study suggests that there is no significant effect on accuracy of expanding the lexical context using WordNet ($Lin=0.1$), even though this results in an increase in rule application of 40%. The study also confirms that there is a sharp and

Rater	FLUENCY			SIMPLICITY			MEANING		
	WN	WI	NC	WN	WI	NC	WN	WI	NC
Mean	3.28	3.59	2.49	3.68	3.51	2.47	2.52	2.72	2.17
SD	1.38	1.31	1.44	1.32	1.28	1.34	1.12	1.11	1.27
Median	4	4	2	3	4	2	3	3	2

Table 4: Results of human evaluation of different versions of simplified text (WN: WordNet-context (Lin=0.1); WI: Wikipedia-context; NC: No-context)

significant drop in accuracy from removing lexical context altogether (the approach used by Wubben et al. (2012), for example). Next, we perform an evaluation of our hybrid text simplification system, that augments the existing RegenT system (Siddharthan, 2011), with its hand-written rules for syntactic simplification, with the automatically acquired lexicalised rules (the Lin=0.1 ruleset).

4 Hybrid text simplification system

The RegenT text simplification toolkit (Siddharthan, 2011) is distributed with a small hand-crafted grammar for common syntactic simplifications: 26 hand-crafted rules for apposition, relative clauses, and combinations of the two; a further 85 rules handle subordination and coordination (these are greater in number because they are lexicalised on the conjunction); 11 further rules cover voice conversion from passive to active; 38 rules for light verbs and various cleft constructions; 99 rules to handle common verbose constructions described in the old GNU diction utility; 14 rules to standardise quotations.

The RegenT system does not have a decoder or a planner. It also does not address discourse issues such as those described in Siddharthan (2003a), though it includes a component that improves relative clause attachment based on Siddharthan (2003b). It applies the simplification rules exhaustively to the dependency parse; i.e., every rule for which the DELETE list is matched is applied iteratively (see Siddharthan (2011) for details).

We have created a hybrid text simplification system by integrating our automatically acquired rules (lexical context extended using WordNet for single change rules, and lexical context removed for longer rules) with the existing RegenT system as described above. This is sensible, as the existing manually written rules for syntactic simplification are more reliable than automatically extracted ones: They model morphological change, allowing for a linguistically accurate treatment of syntactic phenomenon such as voice change. The current work addresses a major limitation

of hand-crafted text simplification systems—such systems restrict themselves to syntactic simplification, even though vocabulary plays a central role in reading comprehension. We hope that the methods described here can extend a hand-crafted system to create a hybrid text simplification system that is accurate as well as wide coverage. We next present a large scale manual evaluation of this hybrid system.

4.1 Evaluation

We performed a manual evaluation of how fluent and simple the text produced by our simplification system is, and the extent to which it preserves meaning.

Our system (henceforth, HYBRID) is compared to QTSG, the system by Woodsend and Lapata (2011) that learns a quasi-synchronous tree substitution grammar. This is the best performing system in the literature with a similar scope to ours in terms of the syntactic and lexical operations performed². Further the two systems are trained on the same data. QTSG relies entirely on an automatically acquired grammar of 1431 rules. Our automatically extracted grammar has 5466 lexicalised rules to augment the existing manually written syntactic rules in RegenT.

We also compare the two systems to the manual gold standard SEW, and against the original EW sentences.

Data: We use the evaluation set previously used by several others (Woodsend and Lapata, 2011; Wubben et al., 2012; Zhu et al., 2010). This consists of 100 sentences from English Wikipedia (EW), aligned with Simple English Wikipedia (SEW) sentences. These 100 sentences have been excluded from our training data for rule acquisition, as is standard. Following the protocol of Wubben et al. (2012), we used all the sentences from the evaluation set for which both QTSG and

²The PBMT system of Wubben et al. (2012) reports better results than QTSG, but is not directly comparable because it does not perform sentence splitting, and also trains on a different corpus of news headlines.

Rater	FLUENCY				SIMPLICITY				MEANING			
	EW	SEW	QTSG	HYB	EW	SEW	QTSG	HYB	EW	SEW	QTSG	HYB
Mean	3.99	4.06	1.97	3.52	3.43	3.58	2.33	3.73	-	4.03	2.23	3.40
SD	0.94	1.00	1.24	1.24	1.07	1.22	1.26	1.30	-	1.02	1.23	1.18
Median	4	4	1	4	3	4	2	4	-	4	2	3

Table 5: Results of human evaluation of different simplified texts (EW: English Wikipedia; SEW: Simple English Wikipedia; QTSG: Woodsend and Lapata (2011) system; HYB: Our hybrid system)

HYBRID had performed at least one simplification (as selecting sentences where no simplification is performed by one system is likely to boost its fluency and meaning preservation ratings). This gave us a test set of 62 sentences from the original 100.

Method: We recruited participants on Amazon Mechanical Turk, filtered to live in the US and have an approval rating of 80%. These participants were shown examples containing the original Wikipedia sentence, followed by QTSG, HYBRID and SEW in a randomised manner. For each such set, they were asked to rate each simplified version for fluency, simplicity and the extent to which it preserved the meaning of the original. Additionally, participants were also asked to rate the fluency and simplicity of the original EW sentence. We used a Likert scale of 1–5, where 1 is totally unusable output, and 5 is output that is perfectly usable. The experiment resulted in obtaining a total of 3669 ratings for 62 sentences involving 76 raters.

Results: The results are shown in Tab. 5. As seen, our HYBRID system outperforms QTSG in all three metrics and is indeed comparable to the SEW version when one looks at the median scores. Interestingly, our system performs better than SEW with respect to simplicity, suggesting that the hybrid system is indeed capable of a wide range of simplification operations. The ANOVA tests carried out to measure significant differences between versions is presented below.

Fluency: A one-way ANOVA was conducted to evaluate fluency for versions of simplified text showed a highly significant effect of version (EW, SEW, QTSG, HYBRID) on the fluency score ($F=695.2$, $p<10^{-16}$). A Tukey’s pairwise comparison test (Tukey’s HSD, overall alpha level = 0.05) indicated significant differences between QTSG-EW; HYBRID-EW; HYBRID-QTSG; SEW-QTSG; SEW-HYBRID at $p = 0.01$.

Simplicity: A one-way ANOVA conducted to evaluate fluency for versions of simplified text showed a highly significant effect of version

(EW, SEW, QTSG, HYBRID) on the simplicity score ($F=29.9$, $p<10^{-16}$). A Tukey’s pairwise comparison test (Tukey’s HSD, overall alpha level = 0.05) indicated significant differences between QTSG-EW; HYBRID-EW; HYBRID-QTSG; SEW-QTSG; all at $p<0.01$.

Meaning Preservation: A one-way ANOVA conducted to evaluate meaning preservation for versions of simplified text showed a highly significant effect of version (EW, SEW, QTSG, HYBRID) on the meaning preservation score ($F=578.1$, $p=2\times 10^{-16}$). A Tukey’s pairwise comparison test (Tukey’s HSD, overall alpha level = 0.05) indicated significant differences between QTSG-SEW; HYBRID-SEW; and HYBRID-QTSG all at $p<0.01$.

5 Conclusion

We have described a hybrid system that performs text simplification using synchronous dependency grammars. The grammar formalism is intuitive enough to write rules by hand, and a syntactic rule set is distributed with the RegenT system. The contributions of this paper are to demonstrate that the same framework can be used to acquire lexicalised rules from a corpus, and that the resultant system generates simplified sentences that are comparable to those written by humans.

We have documented how a grammar can be extracted from a corpus, filtered and generalised. Our studies confirm the benefits of generalising rules in this manner. The resultant system that combines this grammar with the existing manual grammar for syntactic simplification has outperformed the best comparable contemporary system in a large evaluation. Indeed our system performs at a level comparable to the manual gold standard in a substantial evaluation involving 76 participants, suggesting that text simplification systems are reaching maturity for real application.

Acknowledgements

This research is supported by an award made by the EPSRC; award reference: EP/J018805/1.

References

- Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.
- Arnaldo Candido Jr, Erick Maziero, Caroline Gasperin, Thiago AS Pardo, Lucia Specia, and Sandra M Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics.
- Yvonne Canning. 2002. *Syntactic simplification of Text*. Ph.D. thesis, University of Sunderland, UK.
- Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044, Copenhagen, Denmark.
- William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9. Association for Computational Linguistics.
- Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26.
- M.C. De Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454. Citeseer.
- Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. In J. Nerbonne, editor, *Linguistic Databases*, pages 161–173. CSLI Publications, Stanford, California.
- Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 541–548. Association for Computational Linguistics.
- Decang Lin. 1998. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.
- Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*.
- Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Advaith Siddharthan. 2003a. Preserving discourse structure when simplifying text. In *Proceedings of the European Natural Language Generation Workshop (ENLG), 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 103–110, Budapest, Hungary.
- Advaith Siddharthan. 2003b. Resolving pronouns robustly: Plumbing the depths of shallowness. In *Proceedings of the Workshop on Computational Treatments of Anaphora, 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 7–14, Budapest, Hungary.
- Advaith Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proc. of the 6th International Natural Language Generation Conference (INLG 2010)*, pages 125–133. Dublin, Ireland.
- Advaith Siddharthan. 2011. Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 2–11. Association for Computational Linguistics.
- Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 347–355. Association for Computational Linguistics.
- Lucia Specia. 2010. Translating from complex to simplified sentences. In *Proceedings of the Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.
- Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–420. Association for Computational Linguistics.

- Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.
- Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics.
- Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368. Association for Computational Linguistics.
- Zhemín Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.