Lexico-syntactic text simplification and compression with typed dependencies

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

Tadashi Nomoto
National Institute of
Japanese Literature,
Tokyo, Japan.
nomoto@acm.org

Advaith Siddharthan
Computing Science,
University of Aberdeen,
Aberdeen, UK.
advaith@abdn.ac.uk

Abstract

We describe two systems for text simplification using typed dependency structures, one that performs lexical and syntactic simplification, and another that performs sentence compression optimised to satisfy global text constraints such as lexical density, the ratio of difficult words, and text length. We report a substantial evaluation that demonstrates the superiority of our systems, individually and in combination, over the state of the art, and also report a comprehension based evaluation of contemporary automatic text simplification systems with target non-native readers.

1 Introduction

Text simplification has often been defined as the process of reducing the grammatical and lexical complexity of a text, while still retaining the original information content and meaning. However, text can also be simplified in other ways; for instance, by removing peripheral information to reduce text length, through sentence compression or summarisation. A key goal of automatic text simplification is to make information more accessible to the large numbers of people with reduced literacy, motivated by a large body of evidence that manual text simplification is an effective intervention (Anderson and Freebody, 1981; L'Allier, 1980; Beck et al., 1991; Anderson and Davison, 1988; Linderholm et al., 2000; Kamalski et al., 2008). However automatic text simplification systems have rarely been evaluated in a manner that sheds light on whether they can facilitate target users.

To date, evaluations of automatic text simplification have been (a) performed on a small scale, as few as 20–25 sentences in some cases (Wubben et al., 2012; Siddharthan and Mandya, 2014; Narayan and Gardent, 2014), (b) performed on sentences in isolation, thus not measuring incoherence caused at the inter-sentential level that can make text more difficult (Siddharthan (2003a) being the exception), and (c) performed using either automatic metrics (Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Wubben et al., 2012; Paetzold and Specia, 2013) or using ratings by fluent readers for fluency, simplicity and meaning preservation (Siddharthan, 2006; Woodsend and Lapata, 2011; Wubben et al., 2012; Paetzold and Specia, 2013; Siddharthan and Mandya, 2014; Narayan and Gardent, 2014; Mandya and Siddharthan, 2014). As such, none of these evaluations can help us answer the basic question: How good is automatic text simplification; i.e., would it facilitate poor readers?

Our goals in this paper are twofold. First, we want to evaluate text simplification systems more systematically than has been attempted before, using both human judgements on a larger scale, and directly testing comprehension on longer passages for target reader populations. Second, we want to compare two different approaches to text simplification. In this paper, we present a text simplification system that can perform lexical and syntactic simplification (§3), as well as a novel sentence compression system designed specifically for the text simplification task (§4), in that it favours compressions with fewer difficult words and with more function words such as connectives that are known to improve readability. We evaluate both, as well as a hybrid system that performs both text simplification and compression (§5, 6).

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1996–2006, Dublin, Ireland, August 23-29 2014.

2 Related Work

Text simplification systems differ primarily in the level of linguistic knowledge they encode. Phrase Based Machine Translation (PBMT) systems (Specia, 2010; Wubben et al., 2012; Coster and Kauchak, 2011) use the least knowledge, and as such are ill equipped to handle simplifications that require morphological changes, syntactic reordering, sentence splitting or insertions. While syntax based MT approaches use syntactic knowledge, existing systems do not offer a treatment of morphology (Zhu et al., 2010; Woodsend and Lapata, 2011; Paetzold and Specia, 2013). This means that while some syntactic reordering operations can be performed well, others requiring morphological changes cannot. Consider converting passive to active voice (e.g., from "trains are liked by John" to "John likes trains"). Besides deleting auxiliaries and reordering the arguments of the verb, there is also a requirement to modify the verb to make it agree in number with the new subject "John", and take the tense of the auxiliary "are".

Hand crafted systems such as Siddharthan (2010) and Siddharthan (2011) use transformation rules that encode morphological changes as well as deletions, re-orderings, substitutions and sentence splitting, and can handle voice change correctly. However, hand crafted systems are limited in scope to syntactic simplification as there are too many lexico-syntactic and lexical simplifications to enumerate manually.

Some contemporary work in text simplification has evolved from research in sentence compression, a related research area that aims to shorten sentences for the purpose of summarising the main content. Sentence compression has historically been addressed in a generative framework, where transformation rules are learnt from parsed corpora of sentences aligned with manually compressed versions, using ideas adapted from statistical machine translation. The compression rules learnt are typically syntactic tree-to-tree transformations (Knight and Marcu, 2000; Galley and McKeown, 2007; Riezler et al., 2003; Cohn and Lapata, 2009; Nomoto, 2008) of some variety. Indeed, Woodsend and Lapata (2011) develop this line of research. Their model is based on quasi-synchronous tree substitution grammar (QTSG) (Smith and Eisner, 2006) and integer linear programming. Quasi-synchronous grammars aim to relax the isomorphism constraints of synchronous grammars, in this case by generating a loose alignment between parse trees. Woodsend and Lapata (2011) use QTSG to generate all possible rewrite operations for a source tree, and then integer linear programming to select the most appropriate simplification. Their system performs lexical and syntactic simplification as well as compression.

Recently, there have been attempts to combine approaches. Narayan and Gardent (2014) use an approach based on semantics to perform syntactic simplification, and PBMT for lexical simplifications. We have also created a hybrid system, but one using linguistically sound hand written rules for syntactic simplification and automatically acquired rules for lexicalised constructs (Siddharthan and Mandya, 2014; Mandya and Siddharthan, 2014). In this paper we combine this work (summarised in §3) with a new method for sentence compression (described in §4).

3 Text Simplification with Synchronous Dependency Grammars

We use the RegenT text simplification (Siddharthan, 2011), augmented with automatically acquired rules, as described in detail elsewhere (Mandya and Siddharthan, 2014; Siddharthan and Mandya, 2014). In this section, we will restrict ourselves to summarising the key features of the system.

Our text simplification system follows the architecture proposed in Ding and Palmer (2005) for Synchronous Dependency Insertion Grammars, reproduced in Fig. 1. It uses the same dataset[1] as Woodsend and Lapata (2011) for learning lexicalised rules. The rules are acquired in the format required by the RegenT text simplification system (Siddharthan, 2011), which is used to implement the simplification. This

[1] consisting of ∼140K aligned simplified and original sentence pairs obtained from Simple English Wikipedia and English Wikipedia.

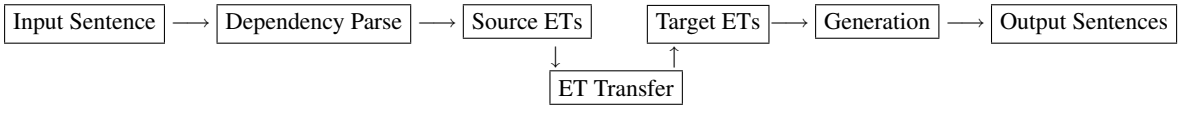

Figure 1: System Architecture

RULE 1: MOST_INTENSIVE2STRONGEST

1. DELETE
   (a) advmod(?X0[intensive], ?X1[most])
   (b) advmod(?X2[storm], ?X0[intensive])
2. INSERT
   (a) advmod(?X2, ?X3[strongest])

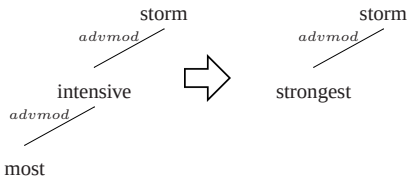

Figure 2: Simplification as a Transfer rule and a transduction of Elementary Trees (ETs)

requires dependency parses from the Stanford Parser, and generates output sentences from dependency parses using the generation-light approach described in (Siddharthan, 2011).

In short, we extract a synchronous grammar from dependency parses of aligned English and simple English sentences, starting from the differences in the parses. 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 reveals that there are two typed dependencies that occur only in the parse of the first sentence, and one that occurs only in the parse of the second. Thus, to convert the first sentence into the second, two dependencies need to be deleted and one inserted. From this example, the rule shown in Fig. 2 is extracted. The rule contains variables (?Xn), 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 handle morphology and to avoid broken dependency links in the Target ETs. These are enumerated in (Siddharthan, 2011). By collecting such rules, a meta-grammar is produced 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". The ET Transfer component can be presented either as transformation rules or as a transduction of ETs, as shown in Fig. 2. In Mandya and Siddharthan (2014), we describe how such automatically acquired rules can be generalised to apply in new contexts; for instance, by expanding lexical context to include related words derived from WordNet, or by removing the lexical context for lexical simplifications that are not context dependent.

Learning paraphrase with typed dependency representations has certain advantages to PBMT; for example, consider the rule that simplifies "described as" to "called":

RULE: DESCRIBED_AS2CALLED

1. DELETE:
   (a) prep_as(?X0[described], ?X1)
2. INSERT:
   (a) dobj(?X2[called], ?X1)

This single rule can simplify "*Coulter was described as a polemicist*" to "*Coulter was called a polemicist*" as well as cases where the words are not adjacent, such as "*Coulter has described herself as a polemicist*" to "*Coulter has called herself a polemicist*".

Our text simplification system, as evaluated in this paper, combines a set of 278 hand crafted grammar for syntactic simplification (from the original RegenT system) and 5172 automatically acquired rules, based on the principles described above.

4 Sentence Compression with Reluctant Trimmer

This section describes the mechanics of the reluctant trimmer (RT), or how it works to create a simplified form of sentence. We will explain later where the word 'reluctant' comes from. Broadly, RT comes in two parts: generation and selection. For a given sentence it takes as input, it generates a number of

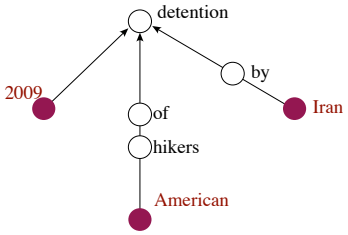

Figure 3: Dependency structure for *"2009 detention of American hikers by Iran"*

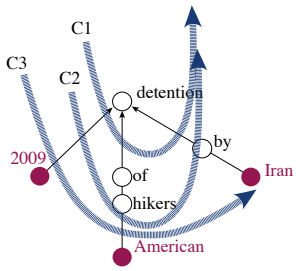

Figure 4: Cropping dependency tree

truncations of the sentence, each of which has some elements removed in a way that largely complies with English syntax. It does this by first parsing the sentence into a a dependency representation, and creating what we call terminating dependency paths out of the representation. After placing them in a lattice format, we run a K-best search over the lattice to generate K best truncations of the sentence. We repeat the process for each sentence found in the text, which will produce a collection of sets of truncation candidates. We then run integer linear programming over the collection, selecting one sentence for each set in a way that satisfies global constraints such as lexical density, the ratio of hard words, and text length. In particular, we regard RT not as an operation that works sentence by sentence, but one that works with text as a whole. We argue that how the sentence is to be compressed is not only dictated by the sentence itself, but also by the text in which it appears.

We start off with an example shown in Figure 3, where we have a phrase "*2009 detention of American hikers by Iran*." Our goal here is to develop a systematic method that will prune the dependency tree so as to generate shorter versions of the sentence largely in compliance with the English grammar. Figure 4 provides an intuitive picture of how this could be done: by cropping the tree along the arrows. We implement the idea by borrowing the notion of *Terminating Dependency Path* (TDP) (Nomoto, 2008), which gives us a way to translate a dependency tree into a trellis of nodes, which in turn allows us to find truncations through dynamic programming.

Figure 5 shows a TDP lattice derived from the dependency tree given in Figure 3. TDPs are depicted as solid blue lines in the figure. It is easy to see that each TDP corresponds to a path in the dependency tree that runs from a leaf to the root. The conversion from dependency tree to TDP lattice is thus straightforward. We perform A* search over the TDP lattice to find the best compression. Assume that we have a path or a sequence of nodes, $\langle n[1], n[2] \ldots, n[j], \ldots, n[z-1], n[z] \rangle$, that takes you from the starting node, $n[1]$, to the goal, $n[z]$, on the TDP lattice. Define the cost C of node $n[x]$ by: $C(x) = g(x) + h(x)$ where $g(x)$ is the cost incurred for the travel from the starting node to $n[x]$ and $h(x)$ the future estimate for the cost of travelling from $n[x]$ to the goal. Let $g(x) = -\sum_{j \in V(1,x)} \text{backward}(j)$ and $h(x) = -\sum_{j \in W(x,z-1)} \text{forward}(j)$, with:

$$\text{backward}(x) = \text{tfidf}(n[x]) + pr(\text{seq}(n[x-1], n[x]) | M), \quad (1)$$

$$\text{forward}(x) = \text{backward}(x+1) \quad (2)$$

$V(1,x)$ is a sequence of nodes that appeared on the path we took to reach $n[x]$ from the starting node, $W(x, z-1)$ a sequence of nodes that gives the shortest possible path (i.e. the path that incurs least cost) from $n[x]$ to the goal. tfidf(n) represents a tfidf score for a word associated with the node n, with tfidf$(n[1]) = 0$ and tfidf$(n[g]) = 0$, and is normalised so that it falls between 1 and 0.[2] seq(n, m) refers to an uninterrupted sequence of words you find on the path that extends from n to m via the root, ignoring duplicates. Figure 6 gives an intuitive sense of how this works. seq (2009, hiker), for instance, can be found by following the blue line in the figure, which results in "*2009 detention of hikers.*" 'M' refers to a language model.[3] $pr(\text{seq}(n, m)|M)$ is the probability of sequence 'seq(n,m)' under language

[2]Document frequencies (df) we used for present purposes are based on those given in the British National Corpus (`www.kilgarriff.co.uk/bnc-readme.html`), which keeps record of the number of files a particular word occurred.

[3]The language model is built here by running SRLM (`www.speech.sri.com/projects/srilm`) on the English

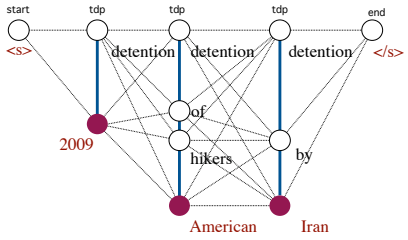

Figure 5: TDP lattice. '$\langle s \rangle$' is a label for the starting node, '$\langle /s \rangle$' that for the goal.

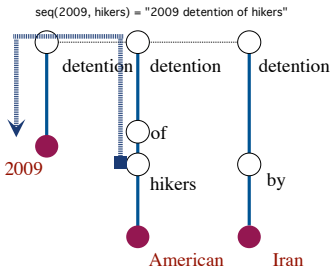

Figure 6: seq(2009,hiker)

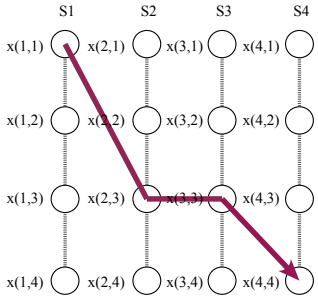

Figure 7: Decoding with ILP

model M.[4] Traversing over the TDP lattice while picking nodes with least costs will produce the best compression, to which we apply Yen (1971)'s algorithm to find K-best alternatives (where K is set to 10 in the experiments reported below).

We now turn to the second part of the story, which is about choosing from each pool of K-best candidates, to create a simplified version of the text. (Recall that we keep a pool of K-best compressions for each of the sentences in a text, and create a simplification by choosing a compression from each pool.) In this paper, we build on a particular simplification approach based on integer linear programming (ILP), by (Dras, 1999), which he dubbed 'reluctant paraphrasing.' In a nutshell, Dras claims that we should make as little change to the text as possible, arguing that any change may run the risk of muddling the meaning of the original text: hence the name 'reluctant paraphrasing.' The following linear program (LP) represents our adaptation of Dras's method. Formula 3 represents the objective function, with 4 through 7 expressing constraints:

$$\min \quad z = \sum c_{i,j} x_{i,j} \qquad (3)$$

subject to:

$$\forall i \sum_j x_{ij} = 1, \quad x_{ij} \in \{0,1\}, \forall ij \qquad (4)$$

$$\frac{W + \sum w_{ij} \cdot x_{ij}}{S} \leq k_1 \qquad (5)$$

$$\frac{H + \sum h_{ij} \cdot x_{ij}}{W + \sum w_{ij} \cdot x_{ij}} \leq k_2 \qquad (6)$$

$$\frac{F + \sum f_{ij} \cdot x_{ij}}{W + \sum w_{ij} \cdot x_{ij}} \geq k_3 \qquad (7)$$

$x_{i,j}$ denotes a candidate for which we are to make a decision on whether to include it in the simplification of a given text d. In particular we mean $x_{i,j}$ to represent the j-th best compression for the i-th sentence in d. Constraint 4 dictates that we have exactly one compression candidate for each sentence in d. w_{ij} indicates the number of changes or deletions we performed on the original sentence to create x_{ij}: -1 if x_{ij} has one less term than the original sentence it is a compression of; 0 if there is no change. W is the number of terms in d, S the number of sentences in d. Constraint 5 states that proportion of the number of terms to that of sentences should be less than or equal to k_1; in other words, changes made to the text should not exceed k_1. H in constraint 6 denotes the total number of 'hard' or difficult words in the original text; h_{ij} the number of changes made to hard words in x_{ij}, namely how many less or more words there remain that are deemed 'hard,' compared to the sentence it comes from.[5]: $h_{ij} = -3$, for example, means that we have three less hard words in x_{ij} than in the original sentence.

Constraint 6 is included here to keep the proportion of hard words in text from growing beyond a

portion of TDT5 corpus and TDT Pilot Study Corpus (both available at Linguistic Data Consortium), the total number of sentences combined reaching 293,971.

[4]We note here that we did not compensate the probability for the length of a word sequence, as we were unable to find an empirical evidence that suggested we should do otherwise.

[5]'Hard words' are defined here as those that fall off of the New General Service List (www.newgeneralservicelist.org) which currently contains 2,881most frequently used words.

particular threshold k_2. The values of k_1, k_2 and k_3 were determined based on the Breaking News English (BNE) corpus (described later), which provides for each story, simplified versions at two levels of difficulty, one being called 'easy' and the other 'hard.' If we take the 'easy' as a gold standard simplification for the 'hard,' we will be able to get estimates of k_1 through k_3. None of the data we used for this purpose, however, is part of the BNE reading test discussed below.

F in constraint 7 represents the total number of function words (those that are not of JJ, MD, NN, RB, or VB in the Penn scheme) while f_{ij} indicates that of changes to function words (the way it works is analogous to h_{ij}). The motivation for the constraint is to prevent function words from being eliminated excessively, which Dras argues, reduces the readability of text. The objective function includes parameters $c_{i,j}$ which serve to indicate the cost of transforming the sentence. In this paper, we define c_{ij} as Levenshtein edit distance between compression and original sentence. In ordinary language, the linear program may read like "Keep changes to a minimum. Accept compressions that look much like the original sentences from which they arise, with less of hard words and content terms and more of function words." Further, we made use of an array of hand-coded constraints in addition to a language model, to ensure that a compression we generate remains as grammatical as possible. Included were those that prohibit the generation of a compression that involves a dangling preposition or breaks apart multi-word prepositions (MWPs) such as *according to*, *compared to*, *in front of*, etc. (the complete list of MWPs we used for this purpose can be found in de Marneffe and Manning (2008)). Added to these were some "don't drop" rules that demanded we keep intact subjects and verbs as well.

Figure 7 illustrates how compression variables $x_{i,j}$ are organised (each of which is depicted as "$x(i, j)$" in the figure). Each vertical line represents a pool of K-best compressions generated for a particular sentence s_i. LP seeks to find a candidate from each pool so that the resulting set of compressions best meets the objective function and conditions it dictates.[6]

5 Evaluation of Fluency, Simplicity and Meaning Preservation

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. We evaluate 3 systems:

TS: The Text Simplification system based on synchronous dependency grammars (§3).

RT: The Reluctant Trimmer for sentence compression (§4).

HYB: A hybrid text simplification system that applies RT to the output of TS.

We used as a baseline Woodsend and Lapata (2011)'s QTSG system that learns a quasi-synchronous tree substitution grammar from the same EW-SEW dataset used by TS. QTSG is the best performing system in the literature with a similar scope to ours in terms of the syntactic, lexical and compression operations performed [7]. QTSG relies entirely on an automatically acquired grammar of 1431 rules, for lexical and syntactic simplification as well as sentence compression. Our TS system has an automatically extracted grammar with 5172 lexicalised rules to augment the existing 278 manually written syntactic rules in RegenT. The RT system is not trained on simplified text. We also compare against the manual simplification (SEW), and the original EW sentences.

Data: We use an evaluation set consisting of 100 sentences from English Wikipedia (EW) aligned with Simple English Wikipedia (SEW) sentences, following recent work (Woodsend and Lapata, 2011; Wubben et al., 2012; Zhu et al., 2010; Mandya and Siddharthan, 2014; Siddharthan and Mandya, 2014). These 100 sentences have been excluded from our training data for rule acquisition, as is standard. Following Wubben et al. (2012), we used all the sentences from the evaluation set for which each of the four systems 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 50 sentences from the original 100.

[6] As an LP solver, we used lp_solve 5.5.2.0, a mixed integer programming solver, available under public license at SourceForge (`lpsolve.sourceforge.net/5.5`).

[7] The PBMT system of Wubben et al. (2012) reports better results than QTSG, but is not directly comparable because it does not perform syntactic simplifications such as sentence splitting.

	FLUENCY						SIMPLICITY						MEANING					
	EW	SEW	QTSG	TS	RT	HYB	EW	SEW	QTSG	TS	RT	HYB	EW	SEW	QTSG	TS	RT	HYB
Mean	3.97	4.09	2.20	3.53	3.19	3.01	3.40	3.54	2.41	3.79	3.15	2.83	-	4.14	2.52	3.44	3.43	3.28
SD	0.92	0.90	1.35	1.12	1.22	1.22	1.08	1.15	1.28	1.18	1.21	1.23	-	0.89	1.31	1.08	1.15	1.14
Median	4	4	2	4	3	3	3	4	2	4	3	3	-	4	2	4	4	3

Table 1: Results of human evaluation of different versions of simplified text

Method: We recruited participants on Amazon Mechanical Turk, filtered to live in the US and have an approval rating of 80%, and paid $3 for a HIT (Human Intelligence Task). Each HIT contained 10 sentences from Wikipedia (EW), each alongside 5 simplified versions: QTSG, TS, RT, HYB and SEW in a randomised manner. For each of these 10 sets, participants were asked to rate each simplified version for fluency, simplicity and the extent to which it preserved the meaning of the original EW sentence. 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.

Results: The results are shown in Table 1. As seen, our HYB system, and the individual components TS and RT all outperform QTSG with all three metrics. In particular, TS is comparable to the SEW version when one looks at the median scores. Interestingly, TS performs better than SEW with respect to simplicity, suggesting that the 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. Table 3 (Row 1) shows the average number of words in the original and each simplified version.

Fluency: A one-way ANOVA was conducted with *fluency* as the dependent variable and text *version* as the fixed effect. We report a significant effect of version (EW, SEW, QTSG, HYB, TS, RT) on the fluency score (F=173.1, p<10^{-16}). A Tukey's pairwise comparison test (Tukey's HSD, overall $\alpha = 0.05$) indicated significant differences between all pairs, except SEW-EW at $p < 0.05$.

Simplicity: A one-way ANOVA was conducted with *simplicity* as the dependent variable and text *version* as the fixed effect. We report a significant effect of version on the simplicity score (F=29.9, p<10^{-16}). A Tukey's pairwise comparison test (Tukey's HSD, overall $\alpha = 0.05$) indicated significant differences between all pairs except: EW-SEW, RT-EW, and SEW-TS at $p < 0.05$.

Meaning: A one-way ANOVA was conducted with *meaning preservation* as the dependent variable and text *version* as the fixed effect. We report a significant effect of version on the meaning preservation score (F=130.12, p=2x10^{-16}). A Tukey's pairwise comparison test (Tukey's HSD, overall $\alpha = 0.05$) indicated significant differences between all pairs except: RT-TS, RT-HYB and HYB-TS at $p < 0.05$.

Error Analysis: We manually examined sentences that had average ratings below 2. The main cause of error for TS was misparsing, particularly errorful relative clause attachment and the parsing of comma separated lists as apposition. TS fails badly in such cases, and it is possible that methods such as those described in Siddharthan (2003b) are still relevant for correcting parser output. RT suffers mainly when it removes punctuation, which make reading difficult, or names that contain meaning (e.g., "*Seven volumes in length , it was composed by Buddhist priest Jien of the Tendai sect c. 1220.*" got compressed to "*Seven volumes in length it was composed by Jien of the sect c. 1220.*"). The hybrid system can create inconsistencies when TS has split a sentence and RT removes names from only one part ("*Moles can be found in most parts of North America, Asia, and Europe, although there are no moles in Ireland.*" got simplified to "*Moles can be found in parts of America, and Asia and Europe. But, there are no moles.*").

6 Evaluation of Reading Comprehension

We also investigate, for the first time, the effect of contemporary text simplification systems on reading comprehension for non-native speakers with a range of English skills.

Method: The test was conducted on Amazon Mechanical Turk with participants chosen from India and paid $0.75 each. There is no method to selectively recruit low reading skill participants on Turk, so these setting were selected to recruit non-native speakers (India) and minimise participants with postgraduate

degrees (low pay). The test comprised of two components - (a) pre-test for English vocabulary skills; and (b) a reading comprehension test to measure the effect of text simplification.

Pre-test: Reading skills are multifaceted and typically assessed through test batteries that test a range of skills. As such there is no comprehensive assessment possible using a single short online test. As we are recruiting non-native speakers, we chose to use the vocabulary size test (Nation and Beglar, 2007), designed to estimate both first language and second language learners' written receptive vocabulary size in English. The test ranks words based on their corpus frequency, and creates 14 levels, each with 1000 words, so that level 10 for example would contain the 9001^{th} to 10000^{th} most frequent words in English. We designed our vocabulary test by using 28 items, 2 at each level[8]. Each word is tested by showing a short sentence containing it and asking the participant to select the meaning of the word from four options. An estimate of vocabulary size can be got by multiplying the score on this test by 500, so the maximum vocabulary size estimate is 28*500=14,000. Nation and Beglar (2007) spell out three important milestones in terms of word family vocabulary size:

> 5000: Minimum for Non-native speakers of non-European backgrounds to cope at English speaking Universities
>
> 8000: Critical goal for language learners to deal with a range of unsimplified language (98% coverage for newspapers)
>
> 9000: Level of non-native English speaking PhD students (98% coverage for English novels)

In addition, we asked participants to self-report their English language skills by selecting from following options: (a) native; (b) fluent (non-native); (c) good (non-native); and (d) basic (non-native).

Main test: The reading comprehension tests were conducted using 5 news summaries chosen from the Breaking News English[9] (BNE) website, with the permission of its creator and maintainer. The BNE website is a resource that provides high quality news summaries at various levels of simplification for second language learners, and has recently been nominated by the British Council for the 2014 ELTons award for Innovation in Learner Resources. We selected five news stories which had manually constructed summaries at reading levels 6 (hard) and 4 (easy). The website provides a range of exercises following each summary at level 6. We chose to use the multiple choice test to assess reading comprehension. For each of these summaries, we created automatically simplified texts by running our systems on the level 6 text. This resulted in a total of five versions for each news summary - L6 (original); L4 (manual simplification); TS (automatic simplification of L6); RT (compression of L6); and HYB (RT applied to output of TS applied to L6).

We used a balanced design where each participant would (after taking the vocabulary pre-test described above) see each of the 5 news stories in exactly one of the 5 versions in a Latin square design. For each comprehension test, the news summary was shown for a maximum of 150 seconds, after which it was removed and 5 multiple choice comprehension questions presented, which was available for another 150 seconds (2.5 minutes). Participants could finish before the 150 seconds by clicking a "finished" button. Table 3 shows the average length of text in each version.

Results: The first row in Table 2 shows the accuracy (proportion of comprehension questions answered correctly) on the main comprehension test for participants divided into four categories based on their estimated vocabulary from the pre-test. We do not find any significant differences, but it appears that the main benefits of automatic text simplification are for moderate readers (vocabulary between 5K and 8K).

We found a very poor correlation between participants' self reported English language skills and their performance on the vocabulary test ($\rho = -0.01$; $p = 0.55$). The poor correlation was due to certain participants over-estimating their skills. Out of 50 participants, 3 rated themselves as native. However, they could get only about 28% of the answers correct, showing the fact that the participants had over-estimated themselves.

This caused us to doubt the reliability of our version of the vocabulary test[10]. We therefore also attempted to categorise participants based on their overall accuracy over all 25 questions in the com-

[8]The original test uses 10 words from each level, but we required a shorter version.

[9]`www.breakingnewsenglish.com`

[10]The published results are for a 140 question test taking 40 minutes, which we have had to reduce to 28 questions for practical reasons.

	L4	L6	TS	RT	HYB	L4	L6	TS	RT	HYB	L4	L6	TS	RT	HYB	L4	L6	TS	RT	HYB
Skills	Excellent (Vocab≥9000)					Good (9000>Vocab≥8000)					Mod (8000>Vocab≥5000)					Poor (Vocab<5000)				
Accuracy	0.69	0.92	0.94	0.85	0.78	0.84	0.87	0.80	0.84	0.77	0.74	0.78	0.80	0.82	0.80	0.64	0.77	0.72	0.58	0.55
Size	13 Participants					10 Participants					14 Participants					13 Participants				
Skills	Excellent (acc≥.9)					Good (.9>acc≥.8)					Mod (.8>acc≥.5)					Poor (acc<.5)				
Accuracy	0.88	0.98	0.95	0.90	0.83	0.75	0.87	0.84	0.82	0.77	0.60	0.70	0.75	0.63	0.58	0.53	0.53	0.40	0.33	0.33
Size	8 Participants					31 Participants					8 Participants					3 Participants				

Table 2: Results of comprehension tests: Mean accuracy (proportion of comprehension questions answered correctly) by reading comprehension skills. Row 1: Participants categorised by estimated vocabulary from pretest. Row 2: Participants categorised based on accuracy on comprehension tests.

	Dataset	Original	Simplified	TS	RT	HYB	QTSG
Average words per text	Wikipedia Evaluation Set	27.0 (EW)	20.4 (SEW)	25.3	22.0	20.6	24.0
Average words per text	Breaking News Evaluation Set	172.6 (L6)	152.8 (L4)	184.4	149.2	151.4	-

Table 3: Effect of simplification of sentence and document lengths

prehension test. While the thresholds of 5000, 8000 and 9000 for vocabulary size are derived from the literature, we had to set these threshold for comprehension scores. To do this in an objective (though still arbitrary) manner, we selected thresholds numerically similar to the vocabulary size thresholds: Excellent ($acc \geq 0.9$), Good($0.9 > acc >= 0.8$), Moderate ($0.8 > acc \geq 0.5$) and Poor ($acc < 0.5$).

The second row in Table 2 shows the accuracy of participants when categorised by average accuracy on the comprehension questions. Note that this categorisation is posthoc (though we have used thresholds derived from the vocabulary test to be objective), and the results pertaining to this categorisation should be regarded as preliminary. This new categorisation based on observed reading ability, rather than predicted language skills, throws up more definitive results. We fitted a Generalised Linear Mixed Model (GLMM), with "correct" answer as the (binary) dependent variable, text "version" (L4, L6, TS, RT, HYB) and "comprehension" (Excellent, Good, Moderate, Poor) as the fixed effects and participant and question as the random effects. We found a strong main effect of comprehension (comprehension=moderate, $z = -3.178$, $p = 0.001$; comprehension=poor, $z = -4.858$, $p < 0.0001$) and a weak effect of version (version=L4, $z = -1.797$, $p = 0.073$); i.e., these three conditions predict a reduced accuracy on the test. We also found a weak interaction between comprehension and version (comprehension=moderate:version=TS, $z = 1.78$, $p = 0.075$); i.e., that TS increases correct answers for readers with moderate reading skills ($p = 0.075$).

Note that L4, RT and HYB all omit information through compression (Table 3 shows text lengths). This explains the drop in comprehension for these versions, as some information needed to answer a question might have been omitted from the summary. Note also that RT and the HYB systems are competitive with the manual simplification L4 for moderate and good readers. Table 4 provides sample texts to illustrate differences.

L6	The United Nations has warned that the Central African Republic (CAR) needs urgent help. The UN Deputy Secretary-General Jan Eliasson said it was 'descending into complete chaos before our eyes'. The landlocked nation has been slowly moving towards a state of total anarchy since rebels seized power in March.
L4	The U.N. has asked for urgent help for the Central African Republic. The UN's Jan Eliasson said it was 'descending into complete chaos'. There is almost a state of anarchy after rebels took power in March.
TS	The United Nations has warned that the Central African Republic , CAR , needs urgent help. The UN Deputy Secretary-General Jan Eliasson said: It was ' descending into complete chaos before our eyes '. The landlocked nation has been slowly moving towards a state of total anarchy. This happened since rebels seized power in March.
RT	The Nations has warned that the Republic needs help. The Deputy Secretary-General Jan Eliasson said it was descending into complete chaos before our eyes. The nation has been slowly moving towards a state of anarchy since rebels seized power in March.
HYB	The Nations has warned that the Central African Republic CAR needs urgent help. The Deputy Secretary-General Jan Eliasson said It was descending into complete chaos before our eyes. The nation has been moving towards a state This happened since rebels seized power.

Table 4: Example of system output to illustrate differences (Beginning of comprehension story 3).

7 Conclusions

We have described and evaluated two different text simplification systems, one that performs lexical and syntactic simplification, and another that performs sentence compression, optimised for the text simplification task. Both systems and their combination outperform a leading contemporary system. The evaluation of reading comprehension with non-native speakers provides preliminary results that automatic text simplification can facilitate comprehension for moderate readers, but not for good ones. A larger evaluation with moderate readers in necessary to confirm this. Finally we plan to make the TS and RT systems available to the public under the Creative Commons license.[11]

Acknowledgements

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

References

Richard C Anderson and Alice Davison. 1988. *Conceptual and empirical bases of readability formulas.* Lawrence Erlbaum Associates, Inc.

Richard Anderson and Peter Freebody. 1981. Vocabulary knowledge. In John Guthrie, editor, *Comprehension and Teaching: Research Reviews*, pages 77–117. International Reading Association, Newark, DE.

Isabel L. Beck, Margaret G. McKeown, Gale M. Sinatra, and Jane A. Loxterman. 1991. Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility. *Reading Research Quarterly*, 26(3):251–276.

T. Cohn and M. Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34(1):637–674.

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.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. http://nlp.stanford.edu/software/dependencies_manual.pdf.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548. Association for Computational Linguistics.

Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Macquarie University NSW 2109 Australia.

Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.

J. Kamalski, T. Sanders, and L. Lentz. 2008. Coherence marking, prior knowledge, and comprehension of informative and persuasive texts: Sorting things out. *Discourse Processes*, 45(4):323–345.

K. Knight and D. Marcu. 2000. Statistics-based summarization — step one: Sentence compression. In *Proceeding of The American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 703–710.

J.J. L'Allier. 1980. *An evaluation study of a computer-based lesson that adjusts reading level by monitoring on task reader characteristics*. Ph.D. thesis, University of Minnesota, Minneapolis, MN.

T. Linderholm, M.G. Everson, P. van den Broek, M. Mischinski, A. Crittenden, and J. Samuels. 2000. Effects of Causal Text Revisions on More-and Less-Skilled Readers' Comprehension of Easy and Difficult Texts. *Cognition and Instruction*, 18(4):525–556.

Angrosh Mandya and Advaith Siddharthan. 2014. Text simplification using synchronous dependency grammars: Generalising automatically harvested rules. In *INLG 2014 Proceedings of the Eighth International Natural Language Generation Conference*, pages 16–25, Philadelphia, PA, June. Association for Computational Linguistics.

[11] For information on the availability of systems, visit us at: `www.quantmedia.org/coling2014/`.

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.*, pages 435–445, Baltimore, MD. Association for Computational Linguistics.

I. S. P. Nation and David Beglar. 2007. A vocabulary size test. *The Language Teacher*, 31(7):9–13.

Tadashi Nomoto. 2008. A generic sentence trimmer with CRFs. In *Proceedings of ACL-08: HLT*, pages 299–307, Columbus, Ohio, June. Association for Computational Linguistics.

Gustavo H. Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 116–125.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, Edmonton, Canada.

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

Advaith Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings 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.

David A Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 23–30. 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.

Jin Y. Yen. 1971. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, July.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.