

Unsupervised Sentence Simplification Using Deep Semantics

Shashi Narayan

School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
snaraya2@inf.ed.ac.uk

Claire Gardent

CNRS, LORIA, UMR 7503
Vandoeuvre-lès-Nancy, F-54500, France
claire.gardent@loria.fr

Abstract

We present a novel approach to sentence simplification which departs from previous work in two main ways. First, it requires neither hand written rules nor a training corpus of aligned standard and simplified sentences. Second, sentence splitting operates on deep semantic structure. We show (i) that the unsupervised framework we propose is competitive with four state-of-the-art supervised systems and (ii) that our semantic based approach allows for a principled and effective handling of sentence splitting.

1 Introduction

Sentence simplification maps a sentence to a simpler, more readable one approximating its content. As has been argued in (Shardlow, 2014), sentence simplification has many potential applications. It is useful as a preprocessing step for a variety of NLP systems such as parsers and machine translation systems (Chandrasekar et al., 1996), summarisation (Knight and Marcu, 2000), sentence fusion (Filippova and Strube, 2008) and semantic role labelling (Vickrey and Koller, 2008). It also has wide ranging potential societal applications as a reading aid for people with aphasia (Carroll et al., 1999), for low literacy readers (Watanabe et al., 2009) and for non native speakers (Siddharthan, 2002).

In this paper, we present a novel approach to sentence simplification which departs from previous work in two main ways. First, it requires neither hand written rules nor a training corpus of aligned standard and simplified sentences. Instead, we exploit non aligned Simple and English Wikipedia to learn the probability of lexical simplifications, of the semantics of simple sentences and of optional phrases i.e., phrase which may be

deleted when simplifying. Second, sentence splitting is semantic based. We show (i) that our unsupervised framework is competitive with four state-of-the-art systems and (ii) that our semantic based approach allows for a principled and effective handling of sentence splitting.

2 Related Work

Earlier work on sentence simplification relied on handcrafted rules to capture syntactic simplification e.g., to split coordinated and subordinated sentences into several, simpler clauses or to model e.g., active/passive transformations (Siddharthan, 2002; Chandrasekar and Srinivas, 1997; Bott et al., 2012; Canning, 2002; Siddharthan, 2011; Siddharthan, 2010). While these hand-crafted approaches can encode precise and linguistically well-informed syntactic transformations, they do not account for lexical simplifications and their interaction with the sentential context. Siddharthan and Mandya (2014) therefore propose an approach where hand-crafted syntactic simplification rules are combined with lexical simplification rules extracted from aligned English and simple English sentences, and revision histories of Simple Wikipedia. They show that such a hybrid symbolic/statistical approach outperforms state of the art systems in a human evaluation based on 25 sentences.

Using the parallel dataset formed by Simple English Wikipedia (SWKP)¹ and traditional English Wikipedia (EWKP)², further work has focused on developing machine learning approaches to sentence simplification.

Zhu et al. (2010) constructed a parallel Wikipedia corpus (PWKP) of 108,016/114,924 complex/simple sentences by aligning sentences from EWKP and SWKP and used the resulting bitext to train a simplification model in-

¹<http://simple.wikipedia.org>

²<http://en.wikipedia.org>

spired by *syntax-based* machine translation (Yamada and Knight, 2001). Their simplification model encodes the probabilities for four rewriting operations on the parse tree of an input sentences namely, substitution, reordering, splitting and deletion. It is combined with a language model to improve grammaticality and the decoder translates sentences into simpler ones by greedily selecting the output sentence with highest probability.

Using both the PWKP corpus developed by Zhu et al. (2010) and the edit history of simple Wikipedia, Woodsend and Lapata (2011) learn a quasi synchronous grammar (Smith and Eisner, 2006) describing a loose alignment between parse trees of complex and of simple sentences. Following Dras (1999), they then generate all possible rewrites for a source tree and use integer linear programming to select the most appropriate simplification. They evaluate their model on the same dataset used by Zhu et al. (2010) namely, an aligned corpus of 100/131 EWKP/SWKP sentences.

Wubben et al. (2012) and Coster and Kauchak (2011) saw simplification as a monolingual translation task where the complex sentence is the source and the simpler one is the target.

To account for deletions, reordering and substitution, Coster and Kauchak (2011) trained a phrase based machine translation system on the PWKP corpus while modifying the word alignment output by GIZA++ in Moses to allow for null phrasal alignments. In this way, they allow for phrases to be deleted during translation.

Similarly, Wubben et al. (2012) used Moses and the PWKP data to train a phrase based machine translation system augmented with a post-hoc reranking procedure designed to rank the output based on their dissimilarity from the source sentence.

Finally, Narayan and Gardent (2014) present a hybrid approach combining a probabilistic model for sentence splitting and deletion with a statistical machine translation system trained on PWKP for substitution and reordering.

Our proposal differs from all these approaches in that it does not use the parallel PWKP corpus for training. Nor do we use hand-written rules. Another difference is that we use a deep semantic representation as input for simplification. While a similar approach was proposed in (Narayan and

Gardent, 2014), the probabilistic models differ in that we determine splitting points based on the maximum likelihood of sequences of thematic role sets present in SWKP whereas Narayan and Gardent (2014) derive the probability of a split from the aligned EWKP/SWKP corpus using expectation maximisation. As we shall see in Section 4, because their data is more sparse, Narayan and Gardent (2014) predicts less and lower quality simplifications by sentence splitting.

3 Simplification Framework

Our simplification framework pipelines three dedicated modules inspired from previous work on lexical simplification, syntactic simplification and sentence compression. All three modules are unsupervised.

3.1 Example Simplification

Before describing the three main modules of our simplification framework, we illustrate its working with an example. Figure 1 shows the input semantic representation associated with sentence (1C) and illustrates the successive simplification steps yielding the intermediate and final simplified sentences shown in (1S₁-S).

- (1) C. In 1964 Peter Higgs published his second paper in Physical Review Letters describing Higgs mechanism which predicted a new massive spin-zero boson for the first time.
S₁ (Lex Simp). In 1964 Peter Higgs wrote his second paper in Physical Review Letters explaining Higgs mechanism which predicted a new massive elementary particle for the first time.
S₂ (Split). In 1964 Peter Higgs wrote his second paper in Physical Review Letters explaining Higgs mechanism. Higgs mechanism predicted a new massive elementary particle for the first time.
S (Deletion). In 1964 Peter Higgs wrote his paper explaining Higgs mechanism. Higgs mechanism predicted a new elementary particle.

First, the input (1C) is rewritten as (1S₁) by replacing standard words with simpler ones using the context aware lexical simplification method proposed in (Biran et al., 2011).

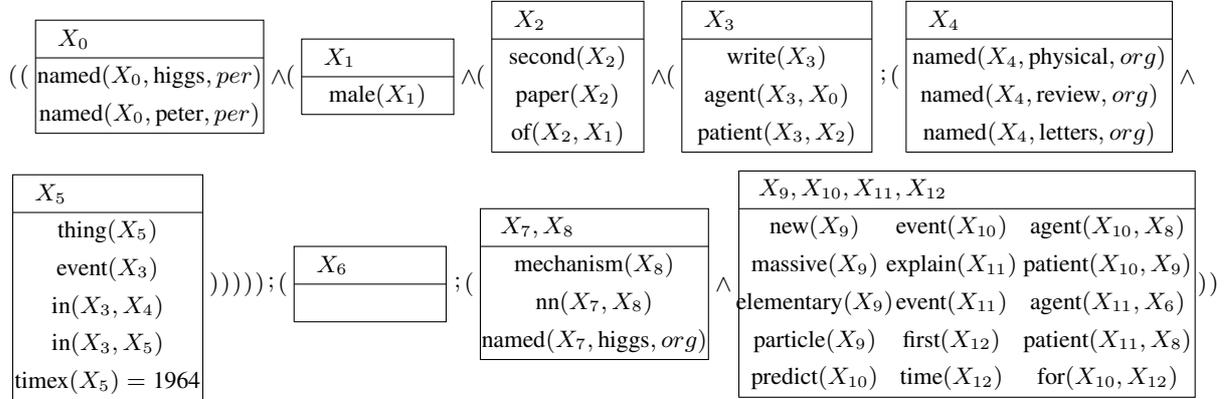
Splitting is then applied to the semantic representation of (1S₁). Following Narayan and Gardent (2014), we use Boxer³ (Curran et al., 2007) to map the output sentence from the lexical simplification step (here 1S₁) to a Discourse Representation Structure (DRS, (Kamp, 1981)). The DRS for (1S₁) is shown at the top of Figure 1 and

³<http://svn.ask.it.usyd.edu.au/trac/candc>, Version 1.00

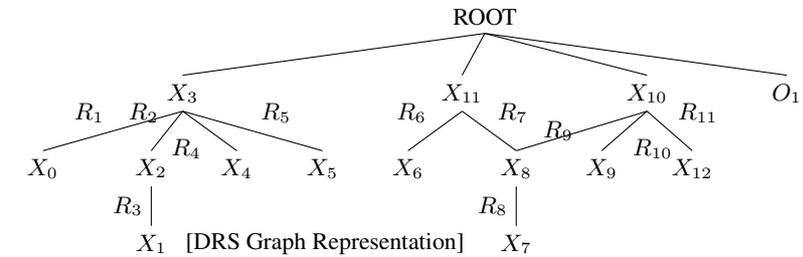
In 1964 Peter Higgs published his second paper in Physical Review Letters describing Higgs mechanism which predicted a new massive spin-zero boson for the first time .

Lex Simpl.

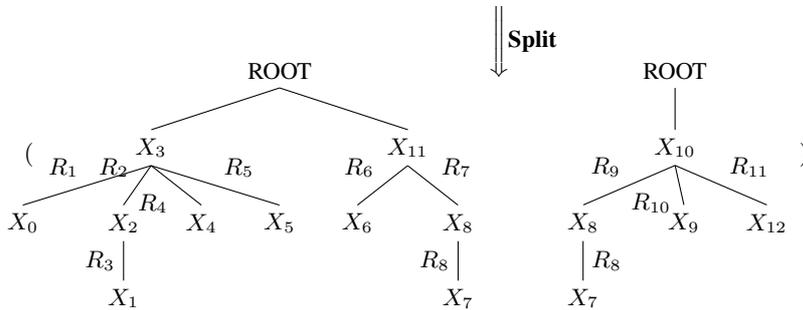
In 1964 Peter Higgs wrote his second paper in Physical Review Letters explaining Higgs mechanism which predicted a new massive elementary particle for the first time .



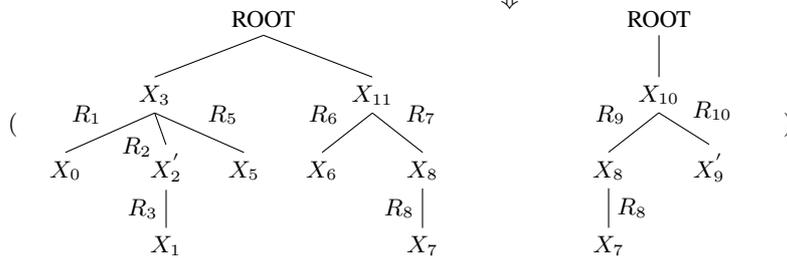
[Discourse Representation Structure produced by BOXER]



node	pos. in S	predicate/type
X ₀	3, 4	higgs/per, peter/per
X ₁	6	male/a
X ₂	6, 7, 8	second/a, paper/n
X ₃	5	write/v, event
X ₄	10, 11, 12	physical/org review/org, letters/org
X ₅	2	thing/n, 1964
X ₆	6, 7, 8	--
X ₇	14	higgs/org
X ₈	14, 15	mechanism/n
X ₉	18, 19, 20 21, 22	new/a, elementary/a massive/a, particle/n
X ₁₀	17	predict/v, event
X ₁₁	13	explain/v, event
X ₁₂	24, 25, 26	first/a, time/n
O ₁	16	which/WDT



Deletion



In 1964 Peter Higgs wrote his paper explaining Higgs mechanism.

Higgs mechanism predicted a new elementary particle.

rel	pos. in S	predicate
R ₁	5	agent, X ₃ → X ₀
R ₂	5	patient, X ₃ → X ₂
R ₃	6	of, X ₂ → X ₁
R ₄	9	in, X ₃ → X ₄
R ₅	1	in, X ₃ → X ₅
R ₆	13	agent, X ₁₁ → X ₆
R ₇	13	patient, X ₁₁ → X ₈
R ₈	--	nn, X ₈ → X ₇
R ₉	17	agent, X ₁₀ → X ₈
R ₁₀	17	patient, X ₁₀ → X ₉
R ₁₁	23	for, X ₁₀ → X ₁₂

Figure 1: Simplification of “In 1964 Peter Higgs published his second paper in Physical Review Letters describing Higgs mechanism which predicted a new massive spin-zero boson for the first time.”

a graph representation⁴ of the dependencies between its variables is shown immediately below. In this graph, each DRS variable labels a node in the graph and each edge is labelled with the relation holding between the variables labelling its end vertices. The two tables to the right of the picture show the predicates (top table) associated with each variable and the relation label (bottom table) associated with each edge. Boxer also outputs the associated positions in the complex sentence for each predicate (not shown in the DRS but shown in the graph tables). Orphan words i.e., words which have no corresponding material in the DRS (e.g., *which* at position 16), are added to the graph (node O_1) thus ensuring that the position set associated with the graph exactly generates the input sentence.

Using probabilities over sequences of thematic role sets acquired from the DRS representations of SWKP, the split module determines where and how to split the input DRS. In this case, one split is applied between X_{11} (*explain*) and X_{10} (*predict*). The simpler sentences resulting from the split are then derived from the DRS using the word order information associated with the predicates, duplicating or pronominalising any shared element (e.g., *Higgs mechanism* in Figure 1) and deleting any Orphan words (e.g., *which*) which occurs at the split boundary. Splitting thus derives $(1S_2)$ from $(1S_1)$.

Finally, deletion or sentence compression applies transforming $(1S_2)$ into $(1S)$.

3.2 Context-Aware Lexical Simplification

We extract context-aware lexical simplification rules from EWKP and SWKP⁵ using the approach described by Biran et al. (2011). The underlying intuition behind these rules is that the word C from EWKP can be replaced with a word S from SWKP if C and S share similar contexts (ten token window) in EWKP and SWKP respectively. Given an input sentence and the set of simplification rules extracted from EWKP and SWKP, we

⁴The DRS to graph conversion goes through several pre-processing steps: the relation *nn* is inverted making modifier noun (*higgs*) dependent of modified noun (*mechanism*), *named* and *timex* are converted to unary predicates, e.g., *named(x, peter)* is mapped to *peter(x)* and *timex(x) = 1964* is mapped to *1964(x)*; and nodes are introduced for orphan words (e.g., *which*).

⁵We downloaded the snapshots of English Wikipedia dated 2013-12-31 and of Simple English Wikipedia dated 2014-01-01 available at <http://dumps.wikimedia.org>.

then consider all possible (C, S) substitutions licensed by the extracted rules and we identify the best combination of lexical simplifications using dynamic programming and rule scores which capture the adequacy, in context, of each possible substitution⁶.

3.3 Sentence Splitting

A distinguishing feature of our approach is that splitting is based on deep semantic representations rather than phrase structure trees – as in (Zhu et al., 2010; Woodsend and Lapata, 2011) – or dependency trees – as in (Siddharthan and Mandya, 2014).

While Woodsend and Lapata (2011) report learning 438 splitting rules for their simplification approach operating on phrase structure trees, Siddharthan and Mandya (2014) defines 26 hand-crafted rules for simplifying apposition and/or relative clauses in dependency structures and 85 rules to handle subordination and coordination.

In contrast, we do not need to specify or to learn complex rewrite rules for splitting a complex sentence into several simpler sentences. Instead, we simply learn the probability of sequences of thematic role sets likely to cooccur in a simplified sentence.

The intuition underlying our approach is that:

Semantic representations give a clear handle on events, on their associated roles sets and on shared elements thereby facilitating both the identification of possible splitting points and the reconstruction of shared elements in the sentences resulting from a split.

For instance, the DRS in Figure 1 makes clear that sentence $(1S_1)$ contains 3 main events and that *Higgs mechanism* is shared between two propositions.

To determine whether and where to split the input sentence, we use a probabilistic model trained on the DRSs of the Simple Wikipedia sentences and a language model also trained on Simple Wikipedia. Given the event variables contained in the DRS of the input sentence, we consider all possible splits between subsequences of events and choose the split(s) with maximum split score. For instance, in the sentence shown in Figure 1,

⁶For more details on the lexical simplification process, we refer the reader to Biran et al. (2011).

there are three event variables X_3 , X_{10} and X_{11} in the DRS. So we will consider 5 split possibilities namely, no split ($\{X_3, X_{10}, X_{11}\}$), two splits resulting in three sentences describing an event each ($\{X_3\}$, $\{X_{10}\}$, $\{X_{11}\}$) and one split resulting in two sentences describing one and two events respectively (i.e., ($\{X_3\}$, $\{X_{10}, X_{11}\}$), ($\{X_3, X_{10}\}$, $\{X_{11}\}$) and $\{X_{10}\}$, $\{X_3, X_{11}\}$). The split $\{X_{10}\}$, $\{X_3, X_{11}\}$ gets the maximum split score and is chosen to split the sentence (1S₁) producing the sentences (1S₂).

Semantic Pattern	prob.
$\langle (agent, patient) \rangle$	0.059
$\langle (agent, in, in, patient) \rangle$	0.002
$\langle (agent, patient), (agent, in, in, patient) \rangle$	0.023

Table 1: Split Feature Table (SFT) showing some of the semantic patterns from Figure 1.

Formally, the split score P_{split} associated with the splitting of a sentence S into a sequence of sentences $s_1 \dots s_n$ is defined as:

$$P_{split} = \sum_{s_i} \frac{1}{n} \times \frac{L_{split}}{L_{split} + |L_{split} - L_{s_i}|} \times lm_{s_i} \times SFT_{s_i}$$

where n is the number of sentences produced after splitting; L_{split} is the average length of the split sentences ($L_{split} = \frac{L_S}{n}$ where L_S is the length of the sentence S); L_{s_i} is the length of the sentence s_i ; lm_{s_i} is the probability of s_i given by the language model and SFT_{s_i} is the likelihood of the semantic pattern associated with s_i . The Split Feature Table (SFT, Table 1) is derived from the corpus of DRSs associated with the SWKP sentences and the counts of sequences of thematic role sets licenced by the DRSs of SWKP sentences. Intuitively, P_{split} favors splits involving frequent semantic patterns (frequent sequences of thematic role sets) and sub-sentences of roughly equal length.

3.4 Phrasal Deletion

Following Filippova and Strube (2008), we formulate phrase deletion as an optimization problem which is solved using integer linear programming⁷. Given the DRS K associated with a sentence to be simplified, for each relation $r \in K$, the deletion module determines whether r and its associated DRS subgraphs should be deleted by maximising the following objective function:

$$\sum_x x_{h,w}^r \times P(r|h) \times P(w) \quad r \notin \{agent, patient, theme, eq\}$$

⁷In our implementation, we use *lp_solve*, <http://sourceforge.net/projects/lpsolve>.

where for each relation $r \in K$, $x_{h,w}^r = 1$ if r is preserved and $x_{h,w}^r = 0$ otherwise; $P(r|h)$ is the conditional probability (estimated on the DRS corpus derived from SWKP) of r given the head label h ; and $P(w)$ is the relative frequency of w in SWKP⁸.

Intuitively, this objective function will favor obligatory dependencies over optional ones and simple words (i.e., words that are frequent in SWKP). In addition, the objective function is subjected to constraints which ensure (i) that some deletion takes place and (ii) that the resulting DRS is a well-formed graph.

4 Evaluation

We evaluate our approach both globally and by module focusing in particular on the splitting component of our simplification approach.

4.1 Global evaluation

The testset provided by Zhu et al. (2010) was used by four supervised systems for automatic evaluation using metrics such as BLEU, sentence length and number of edits. In addition, most recent simplification approaches carry out a human evaluation on a small set of randomly selected complex/simple sentence pairs. Thus Wubben et al. (2012), Narayan and Gardent (2014) and Sidharthan and Mandya (2014) carry out a human evaluation on 20, 20 and 25 sentences respectively.

Accordingly, we perform an automatic comparative evaluation using Zhu et al.’s testset namely, an aligned corpus of 100/131 EWKP/SWKP sentences; and we carry out a human-based evaluation.

Automatic Evaluation Following Wubben et al. (2012), Zhu et al. (2010) and Woodsend and Lapata (2011), we use metrics that are directly related to the simplification task namely, the number of splits in the overall data, the number of output sentences with no edits (i.e., sentences which have not been simplified) and the average Levenshtein distance (LD) between the system output and both the complex and the simple reference sentences. We

⁸To account for modifiers which are represented as predicates on nodes rather than relations, we preprocess the DRSs and transform each of these predicates into a single node subtree of the node it modifies. For example in Figure 1, the node X_2 labeled with the modifier predicate *second* is updated to a new node X'_2 dominating a child labeled with that predicate and related to X_2 by a modifier relation.

System	Levenshtein Edit distance				BLEU w.r.t simple	Sentences with splits	Average sentence length	Average token length
	Complex to System		System to Sim- ple					
	LD	No edit	LD	No edit				
GOLD	12.24	3	0	100	100	28	27.80	4.40
Zhu	7.87	2	14.64	0	37.4	80	24.21	4.38
Woodsend	8.63	24	16.03	2	42	63	28.10	4.50
Wubben	3.33	6	13.57	2	41.4	1	28.25	4.41
Narayan	6.32	4	11.53	3	53.6	10	26.24	4.36
UNSUP	6.75	3	14.29	0	38.47	49	26.22	4.40

Table 2: Automatic evaluation results. Zhu, Woodsend, Wubben, Narayan are the best output of the models of Zhu et al. (2010), Woodsend and Lapata (2011), Wubben et al. (2012) and Narayan and Gardent (2014) respectively; UNSUP is our model.

System	Levenshtein Edit distance				BLEU Scores with respect to		Average sentence length	Average token length
	Complex to System		System to Sim- ple		complex	simple		
	LD	No edit	LD	No edit				
complex	0	100	12.24	3	100	49.85	27.80	4.62
LexSimpl	2.07	22	13.00	1	82.05	44.29	27.80	4.46
Split	2.27	51	13.62	1	89.70	46.15	29.10	4.63
Deletion	2.39	4	12.34	0	85.15	47.33	25.41	4.54
LexSimpl-Split	4.43	11	14.39	0	73.20	41.18	29.15	4.48
LexSimpl-Deletion	4.29	3	13.09	0	69.84	41.91	25.42	4.38
Split-Deletion	4.63	4	13.42	0	77.82	43.44	26.19	4.55
LexSimpl-Split-Deletion	6.75	3	14.29	0	63.41	38.47	26.22	4.40
GOLD (simple)	12.24	3	0	100	49.85	100	23.38	4.40

Table 3: Automated Metrics for Simplification: Modular evaluation. LexSimpl-Split-Deletion is our final system UNSUP.

use BLEU⁹ as a means to evaluate how close the systems output are to the reference corpus.

Table 2 shows the results of the automatic evaluation. The most noticeable result is that our unsupervised system yields results that are similar to those of the supervised approaches.

The results also show that, in contrast to Woodsend system which often leaves the input unsimplified (24% of the input), our system almost always modifies the input sentence (only 3% of the input are not simplified); and that the number of simplifications including a split is relatively high (49% of the cases) suggesting a good ability to split complex sentences into simpler ones.

Human Evaluation Human judges were asked to rate input/output pairs w.r.t. to adequacy (How much does the simplified sentence(s) preserve the meaning of the input?), to simplification (How much does the generated sentence(s) simplify the complex input?) and to fluency (how grammatical and fluent are the sentences?).

We randomly selected 18 complex sentences from Zhu’s test corpus and included in the evaluation corpus: the corresponding simple (Gold) sentence from Zhu’s test corpus, the output of our system (UNSUP) and the output of the other four systems (Zhu, Woodsend, Narayan and Wubben)

which were provided to us by the system authors¹⁰. We collected ratings from 18 participants. All were either native speakers or proficient in English, having taken part in a Master taught in English or lived in an English speaking country for an extended period of time. The evaluation was done online using the LG-Eval toolkit (Kow and Belz, 2012)¹¹ and a Latin Square Experimental Design (LSED) was used to ensure a fair distribution of the systems and the data across raters.

Systems	Simplicity	Fluency	Adequacy
GOLD	3.62	4.69	3.80
Zhu	2.62	2.56	2.47
Woodsend	1.69	3.15	3.15
Wubben	1.52	3.05	3.38
Narayan	2.30	3.03	3.35
UNSUP	2.83	3.56	2.83

Table 4: Average Human Ratings for simplicity, fluency and adequacy.

Table 4 shows the average ratings of the human evaluation on a scale from 0 to 5. Pairwise comparisons between all models and their statistical significance were carried out using a one-way ANOVA with post-hoc Tukey HSD tests. If we group together systems for which there is no significant difference (significance level: $p < 0.05$), our system is in the first group together

⁹Moses support tools: multi-bleu <http://www.statmt.org/moses/?n=Moses.SupportTools>.

¹⁰We upload the outputs from all the systems as supplementary material with this paper.

¹¹<http://www.nltg.brighton.ac.uk/research/lg-eval/>

System pairs		Average Score (number of split sentences)					
A	B	ALL-A	ALL-B	ONLY-A	BOTH-AB		ONLY-B
					A	B	
	GOLD		3.85(28)	2.15(32)	2.80(17)	3.70(17)	4.05(11)
	Zhu		2.25(80)	1.53(4)	2.45(45)	2.42(45)	2.02(35)
UNSUP	Woodsend	2.37(49)	2.08(63)	2.42(11)	2.36(38)	2.29(38)	1.78(25)
	Wubben		2.73(1)	2.32(48)	4.75(1)	2.73(1)	0(0)
	Narayan		2.09(10)	2.29(41)	2.78(8)	1.79(8)	3.81(2)

Table 5: Pairwise split evaluation: Each row shows the pairwise comparison of the quality of splits in UNSUP and some other system. Last six columns show the average scores and number of associated split sentences. The second column (ALL-A) and the third column (ALL-B) present the quality of all splits by systems A and B respectively. The fourth column (ONLY-A) represents sentences where A splits but not B. The fifth and sixth columns represent sentences where both A and B split. The seventh column (ONLY-B) represents sentences where B splits but not A.

with Narayan and Zhu for simplicity; in the first group for fluency; and in the second group for adequacy (together with Woodsend and Zhu). A manual examination of the results indicates that UNSUP achieves good simplicity rates through both deletion and sentence splitting. Indeed, the average word length of simplified sentences is smaller for UNSUP (26.22) than for Wubben (28.25) and Woodsend (28.10); comparable with Narayan (26.19) and higher only than Zhu (24.21).

4.2 Modular Evaluation

To assess the relative impact of each module (lexical simplification, deletion and sentence splitting), we also conduct an automated evaluation on each module separately. The results are shown in Table 3.

One first observation is that each module has an impact on simplification. Thus the average Levenshtein Edit distance (LD) to the source clause (complex) is never null for any module while the number of “No edit” indicates that lexical simplification modifies the input sentence in 78%, sentence splitting 49% and deletion 96% of the cases.

In terms of output quality and in particular, similarity with respect to the target clause, deletion is the most effective (smallest LD, best BLEU score w.r.t. target). Further, the results for average token length indicate that lexical simplification is effective in producing shorter words (smaller average length for this module compared to the other two modules).

Predictably, combining modules yields systems that have stronger impact on the source clause (higher LD to complex, lower number of No Edits) with the full system (i.e., the system combining the 3 modules) showing the largest LD to the sources (LD to complex) and the smallest number of source sentences without simplification (3 No Edits).

4.3 Sentence Splitting Using Deep Semantics

To compare our sentence splitting approach with existing systems, we collected in a second human evaluation, all the outputs for which at least one system applied sentence splitting. The raters were then asked to compare pairs of split sentences produced by two distinct systems and to evaluate the quality (0:very bad to 5:very good) of these split sentences taking into account boundary choice, sentence completion and sentence reordering.

Table 5 shows the results of this second evaluation. For each system pair comparing UNSUP (A) with another system (B), the Table gives the scores and the number of splits of both systems: for the inputs on which both systems split (BOTH-AB), on which only UNSUP splits (ONLY-A) and on which only the compared system split (ONLY-B).

UNSUP achieves a better average score (ALL-A = 2.37) than all other systems (ALL-B column) except Wubben (2.73). However Wubben only achieves one split and on that sentence, UNSUP score is 4.75 while Wubben has a score of 2.73 and produces an incorrect split (cf. S_3 in Figure 6). UNSUP

In terms of numbers of splits, three systems often simplify by splitting namely Zhu (80 splits), Woodsend (63) and UNSUP (49). Interestingly, Narayan, trained on the parallel corpus of Wikipedia and Simplified Wikipedia splits less often (10 splits vs 49 for UNSUP) and less well (2.09 average score versus 2.37 for UNSUP). This is unsurprising as the proportion of splits in SWKP was reported in (Narayan and Gardent, 2014) to be a low 6%. In contrast, the set of observations we use to learn the splitting probability is the set of all sequences of thematic role sets derived from the DRSs of the SWKP corpus.

In sum, the unsupervised, semantic-based splitting strategy allows for a high number (49%) of good quality (2.37 score) sentence splits. Be-

S1	<p>Complex. This array distributes data across multiple disks, but the array is seen by the computer user and operating system as one single disk.</p> <p>Zhu. This array sells data across multiple disks but the array is seen. The computer user and operating as one disk. [0]</p> <p>Woodsend. This array distributes data across several disks. The array is seen by the computer user and operating system as one single disk. [4.85]</p> <p>UNSUP. This array moves data across disks. The array is seen by the computer user and operating system as one disk. [4.83]</p>
S2	<p>Complex. Here, simulations of evolution using evolutionary algorithms and artificial life started with the work of Nils Aall Barricelli in the 1960s, and was extended by Alex Fraser, who published a series of papers on simulation of artificial selection.</p> <p>Zhu. Here, simulations of evolution using evolutionary algorithms and artificial life started with the work of nils aall barricelli in the 1960s, and was extended by alex fraser. Alex fraser published a series of papers on feeling of artificial selection. [3.6]</p> <p>Woodsend. It has Genetic algorithms. Evolution is using artificial life. This is Nils Aall Barricelli in the 1960s. Here, simulations started with the work. He was extended by Alex Fraser. He published a group of papers on simulation of artificial selection. [0.33]</p> <p>UNSUP. Here, models of evolution using evolutionary algorithms and artificial life started with the work in the 1960s, and was led by Alex Fraser. Alex Fraser wrote a series of papers on model of selection. [5]</p>
S3	<p>Complex. By 1928, the regional government was moved from the old Cossack capital Novochoerkassk to Rostov, which also engulfed the nearby Armenian town of Nor Nakhijevan.</p> <p>Zhu. By 1928, the government was moved from the old cossack capital novochoerkassk to rostov. Rostov also of the city the nearby armenian town of nor nakhijevan. [2.8]</p> <p>Woodsend. By 1928, the regional government was moved from the old Cossack capital Novochoerkassk to Rostov. Both also engulfed the nearby Armenian town of Nor Nakhijevan. [3]</p> <p>Wubben. by 1928 , the regional government was moved from the old cossack capital novochoerkassk to rostov. the nearby armenian town of nor nakhichevan. [2.7]</p> <p>Narayan. by 1928, the regional government was moved from the old cossack capital novochoerkassk to rostov. rostov that engulfed the nearby armenian town of nor nakhichevan. [2.7]</p> <p>UNSUP. The regional government was moved from the old Cossack capital Novochoerkassk to Rostov. Rostov also absorbed the nearby town of Nor Nakhijevan. [4.75]</p>

Table 6: Example Outputs for Sentence splitting with their average human annotation scores.

cause there are less possible patterns of thematic role sets in simple sentences than possible configurations of parse/dependency trees for complex sentences, it is less prone to data sparsity than the syntax based approach. Because the probabilities learned are not tied to specific syntactic structures but to more abstract semantic patterns, it is also perhaps less sensitive to parse errors.

4.4 Examples from the Test Set

Table 6 shows some examples from the evaluation dataset which were selected to illustrate the workings of our approach and to help interpret the results in Table 2, 4 and 5.

S1 and S2 and S3 show examples of context-aware unsupervised lexical substitutions which are nicely performed by our system. In S1, *The array distributes data* is correctly simplified to *The array moves data* whereas Zhu’s system incorrectly simplifies this clause to *The array sells data*. Similarly, in S2, our system correctly simplifies *Papers on simulation of artificial selection* to *Papers on models of selection* while the other systems either do not simplify or simplify to *Papers on feeling of selection*

For splitting, the examples show two types of splitting performed by our approach namely, split-

ting of coordinated sentences (S1) and splitting between a main and a relative clause (S2,S3). S2 illustrates how the Woodsend system over-splits, an issue already noticed in (Siddharthan and Mandya, 2014); and how Zhu’s system predicts an incorrect split between a verb (*seen*) and its agent argument (*by the user*). Barring a parse error, such incorrect splits will not be predicted by our approach since, in our cases, splits only occur between (verbalisations of) events. S1, S2 and S3 also illustrates how our semantic based approach allows for an adequate reconstruction of shared elements.

5 Conclusion

A major limitation for supervised simplification systems is the limited amount of available parallel standard/simplified data. In this paper, we have shown that it is possible to take an unsupervised approach to sentence simplification which requires a large corpus of standard and simplified language but no alignment between the two. This allowed for the implementation of contextually aware substitution module; and for a simple, linguistically principled account of sentence splitting and shared element reconstruction.

References

- Or Biran, Samuel Brody, and Noémie 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: short papers-Volume 2*, pages 496–501. Association for Computational Linguistics.
- Stefan Bott, Horacio Saggin, and Simon Mille. 2012. Text simplification tools for spanish. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 1665–1671.
- Yvonne Margaret Canning. 2002. *Syntactic simplification of Text*. Ph.D. thesis, University of Sunderland.
- John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, volume 99, pages 269–270. Cite-seer.
- Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183–190.
- 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)*, pages 1041–1044. Association for Computational Linguistics.
- 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.
- James R Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL) on Interactive Poster and Demonstration Sessions*, pages 33–36. Association for Computational Linguistics.
- Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Macquarie University NSW 2109 Australia.
- Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*, pages 25–32. Association for Computational Linguistics.
- Hans Kamp. 1981. A theory of truth and semantic representation. In J.A.G. Groenendijk, T.M.V. Janssen, B.J. Stokhof, and M.J.B. Stokhof, editors, *Formal methods in the study of language*, number pt. 1 in Mathematical Centre tracts. Mathematisch Centrum.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI) and Twelfth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 703–710. AAAI Press.
- Eric Kow and Anja Belz. 2012. LG-Eval: A Toolkit for Creating Online Language Evaluation Experiments. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 4033–4037.
- Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Natural Language Processing*.
- 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. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference (LEC)*, pages 64–71. IEEE Computer Society.
- Advaith Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings of the 6th International Natural Language Generation Conference (INLG)*, pages 125–133. Association for Computational Linguistics.
- 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 (ENLG)*, 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 HLT-NAACL Workshop on Statistical Machine Translation*, pages 23–30. Association for Computational Linguistics.
- David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL) and the Human Language Technology Conference (HLT)*, pages 344–352.

- Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Matos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36. ACM.
- 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 (EMNLP)*, 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 (ACL): 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 (ACL)*, pages 523–530. Association for Computational Linguistics.
- 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 (COLING)*, pages 1353–1361, Stroudsburg, PA, USA. Association for Computational Linguistics.