Hybrid Simplification using Deep Semantics and Machine Translation

Shashi Narayan
Université de Lorraine, LORIA
Villers-lès-Nancy, F-54600, France
shashi.narayan@loria.fr

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

Abstract

We present a hybrid approach to sentence simplification which combines deep semantics and monolingual machine translation to derive simple sentences from complex ones. The approach differs from previous work in two main ways. First, it is semantic based in that it takes as input a deep semantic representation rather than e.g., a sentence or a parse tree. Second, it combines a simplification model for splitting and deletion with a monolingual translation model for phrase substitution and reordering. When compared against current state of the art methods, our model yields significantly simpler output that is both grammatical and meaning preserving.

1 Introduction

Sentence simplification maps a sentence to a simpler, more readable one approximating its content. Typically, a simplified sentence differs from a complex one in that it involves simpler, more usual and often shorter, words (e.g., *use* instead of *exploit*); simpler syntactic constructions (e.g., no relative clauses or apposition); and fewer modifiers (e.g., *He slept* vs. *He also slept*). In practice, simplification is thus often modeled using four main operations: *splitting* a complex sentence into several simpler sentences; *dropping* and *reordering* phrases or constituents; *substituting* words/phrases with simpler ones.

As has been argued in previous work, 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 application as a reading aid for people with aphasis (Carroll et al., 1999), for low literacy readers (Watanabe et al., 2009) and for non native speakers (Siddharthan, 2002).

There has been much work recently on developing computational frameworks for sentence simplification. Synchronous grammars have been used in combination with linear integer programming to generate and rank all possible rewrites of an input sentence (Dras, 1999; Woodsend and Lapata, 2011). Machine Translation systems have been adapted to translate complex sentences into simple ones (Zhu et al., 2010; Wubben et al., 2012; Coster and Kauchak, 2011). And handcrafted rules have been proposed to model the syntactic transformations involved in simplifications (Siddharthan et al., 2004; Siddharthan, 2011; Chandrasekar et al., 1996).

In this paper, we present a hybrid approach to sentence simplification which departs from this previous work in two main ways.

First, it combines a model encoding probabilities for splitting and deletion with a monolingual machine translation module which handles reordering and substitution. In this way, we exploit the ability of statistical machine translation (SMT) systems to capture phrasal/lexical substitution and reordering while relying on a dedicated probabilistic module to capture the splitting and deletion operations which are less well (deletion) or not at all (splitting) captured by SMT approaches.

Second, our approach is semantic based. While previous simplification approaches starts from either the input sentence or its parse tree, our model takes as input a deep semantic representation namely, the Discourse Representation Structure (DRS, (Kamp, 1981)) assigned by Boxer (Curran et al., 2007) to the input complex sentence. As we shall see in Section 4, this permits a linguistically principled account of the splitting operation in that semantically shared elements are taken to be the basis for splitting a complex sentence into several simpler ones; this facilitates completion (the re-creation of the shared element in the split sentences); and this provide a natural means to avoid deleting obligatory arguments.

When compared against current state of the art methods (Zhu et al., 2010; Woodsend and Lapata, 2011; Wubben et al., 2012), our model yields significantly simpler output that is both grammatical and meaning preserving.

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 active/passive transformations (Siddharthan, 2002; Chandrasekar and Srinivas, 1997; Bott et al., 2012; Canning, 2002; Siddharthan, 2011; Siddharthan, 2010). While these handcrafted approaches can encode precise and linguistically well-informed syntactic transformation (using e.g., detailed morphological and syntactic information), they are limited in scope to purely syntactic rules and do not account for lexical simplifications and their interaction with the sentential context.

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

Zhu et al. (2010) constructed a parallel 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 inspired 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 and show that they achieve better BLEU score. They also conducted a human evaluation on 64 of the 100 test sentences and showed again a better performance in terms of simplicity, grammaticality and meaning preservation.

In (Wubben et al., 2012; Coster and Kauchak, 2011), simplification is viewed 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. No human evaluation is provided but the approach is shown to result in statistically significant improvements over a traditional phrase based approach. Similarly, Wubben et al. (2012) use 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. A human evaluation on 20 sentences randomly selected from the test data indicates that, in terms of fluency and adequacy, their system is judged to outperform both Zhu et al. (2010) and Woodsend and Lapata (2011) systems.

3 Simplification Framework

We start by motivating our approach and explaining how it relates to previous proposals w.r.t., the four main operations involved in simplification namely, splitting, deletion, substitution and reordering. We then introduce our framework.

[1] SWKP (`http://simple.wikipedia.org`) is a corpus of simple texts targeting “children and adults who are learning English Language” and whose authors are requested to “use easy words and short sentences”.

[2] `http://en.wikipedia.org`

Sentence Splitting. Sentence splitting is arguably semantic based in that in many cases, splitting occurs when the same semantic entity participates in two distinct eventualities. For instance, in example (1) below, the split is on the noun *bricks* which is involved in two eventualities namely, *"being resistant to cold"* and *"enabling the construction of permanent buildings"*.

(1) **C.** Being more resistant to cold, bricks enabled the construction of permanent buildings.
S. Bricks were more resistant to cold. Bricks enabled the construction of permanent buildings.

While splitting opportunities have a clear counterpart in syntax (i.e., splitting often occurs whenever a relative, a subordinate or an appositive clause occurs in the complex sentence), completion i.e., the reconstruction of the shared element in the second simpler clause, is arguably semantically governed in that the reconstructed element corefers with its matching phrase in the first simpler clause. While our semantic based approach naturally accounts for this by copying the phrase corresponding to the shared entity in both phrases, syntax based approach such as Zhu et al. (2010) and Woodsend and Lapata (2011) will often fail to appropriately reconstruct the shared phrase and introduce agreement mismatches because the alignment or rules they learn are based on syntax alone. For instance, in example (2), Zhu et al. (2010) fails to copy the shared argument *"The judge"* to the second clause whereas Woodsend and Lapata (2011) learns a synchronous rule matching (VP and VP) to (VP. NP(It) VP) thereby failing to produce the correct subject pronoun (*"he"* or *"she"*) for the antecedent *"The judge"*.

(2) **C.** The judge ordered that Chapman should receive psychiatric treatment in prison and sentenced him to twenty years to life.
S_1. The judge ordered that Chapman should get psychiatric treatment. In prison and sentenced him to twenty years to life. (Zhu et al., 2010)
S_2. The judge ordered that Chapman should receive psychiatric treatment in prison. It sentenced him to twenty years to life. (Woodsend and Lapata, 2011)

Deletion. By handling deletion using a probabilistic model trained on semantic representations, we can avoid deleting obligatory arguments. Thus in our approach, semantic subformulae which are related to a predicate by a core thematic roles (e.g., *agent* and *patient*) are never considered for deletion. By contrast, syntax based approaches (Zhu et al., 2010; Woodsend and Lapata, 2011) do not distinguish between optional and obligatory arguments. For instance Zhu et al. (2010) simplifies (3C) to (3S) thereby incorrectly deleting the obligatory theme (*gifts*) of the complex sentence and modifying its meaning to *giving knights and warriors* (instead of *giving gifts to knights and warriors*).

(3) **C.** Women would also often give knights and warriors gifts that included thyme leaves as it was believed to bring courage to the bearer.
S. Women also often give knights and warriors. Gifts included thyme leaves as it was thought to bring courage to the saint. (Zhu et al., 2010)

We also depart from Coster and Kauchak (2011) who rely on null phrasal alignments for deletion during phrase based machine translation. In their approach, deletion is constrained by the training data and the possible alignments, independent of any linguistic knowledge.

Substitution and Reordering SMT based approaches to paraphrasing (Barzilay and Elhadad, 2003; Bannard and Callison-Burch, 2005) and to sentence simplification (Wubben et al., 2012) have shown that by utilising knowledge about alignment and translation probabilities, SMT systems can account for the substitutions and the reorderings occurring in sentence simplification. Following on these approaches, we therefore rely on phrase based SMT to learn substitutions and reordering. In addition, the language model we integrate in the SMT module helps ensuring better fluency and grammaticality.

3.1 An Example

Figure 1 shows how our approach simplifies (4C) into (4S).

(4) **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.
S. Peter Higgs wrote his paper explaining Higgs mechanism in 1964. Higgs mechanism predicted a new elementary particle.

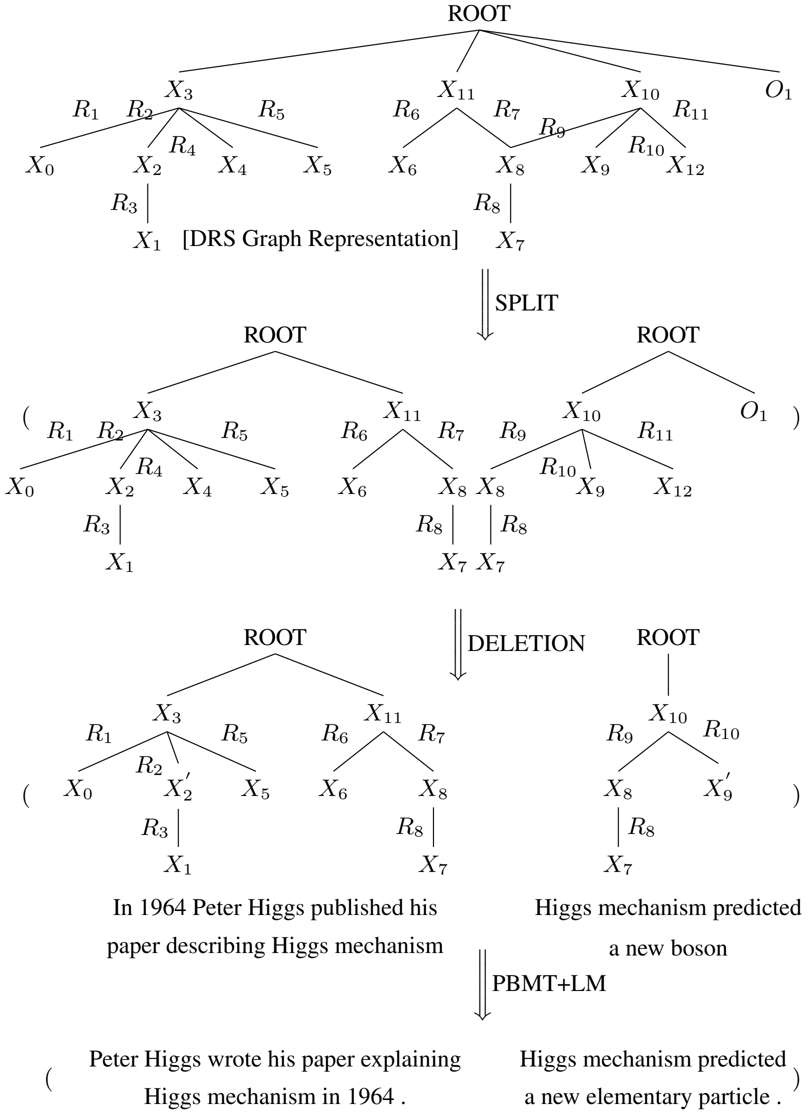
The DRS for (4C) produced using Boxer (Curran et al., 2007) is shown at the top of the Figure and a graph representation[3] of the dependencies between its variables is shown immediately below. Each DRS variable labels a node in the graph and each edge is labelled with the relation holding between the variables labelling its end vertices. The

[3] The DRS to graph conversion goes through several preprocessing 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*).

$$((\begin{array}{|c|}\hline X_0\\\hline \text{named}(X_0,\text{higgs},per)\\ \text{named}(X_0,\text{peter},per)\\\hline\end{array}\wedge(\begin{array}{|c|}\hline X_1\\\hline \text{male}(X_1)\\\hline\end{array}\wedge(\begin{array}{|c|}\hline X_2\\\hline \text{second}(X_2)\\ \text{paper}(X_2)\\ \text{of}(X_2,X_1)\\\hline\end{array}\wedge(\begin{array}{|c|}\hline X_3\\\hline \text{publish}(X_3)\\ \text{agent}(X_3,X_0)\\ \text{patient}(X_3,X_2)\\\hline\end{array};(\begin{array}{|c|}\hline X_4\\\hline \text{named}(X_4,\text{physical},org)\\ \text{named}(X_4,\text{review},org)\\ \text{named}(X_4,\text{letters},org)\\\hline\end{array}\wedge$$

$$\begin{array}{|c|}\hline X_5\\\hline \text{thing}(X_5)\\ \text{event}(X_3)\\ \text{in}(X_3,X_4)\\ \text{in}(X_3,X_5)\\ \text{timex}(X_5)=1964\\\hline\end{array})))));(\begin{array}{|c|}\hline X_6\\\hline \\\hline\end{array};(\begin{array}{|c|}\hline X_7,X_8\\\hline \text{mechanism}(X_8)\\ \text{nn}(X_7,X_8)\\ \text{named}(X_7,\text{higgs},org)\\\hline\end{array}\wedge\begin{array}{|ccc|}\hline X_9,X_{10},X_{11},X_{12} & & \\\hline \text{new}(X_9) & \text{event}(X_{10}) & \text{agent}(X_{10},X_8)\\ \text{massive}(X_9) & \text{describe}(X_{11}) & \text{patient}(X_{10},X_9)\\ \text{spin-zero}(X_9) & \text{event}(X_{11}) & \text{agent}(X_{11},X_6)\\ \text{boson}(X_9) & \text{first}(X_{12}) & \text{patient}(X_{11},X_8)\\ \text{predict}(X_{10}) & \text{time}(X_{12}) & \text{for}(X_{10},X_{12})\\\hline\end{array}$$

[Discourse Representation Structure produced by BOXER]


ROOT
X_3 X_{11} X_{10} O_1
R_1 R_2 R_4 R_5 R_6 R_7 R_9 R_{10} R_{11}
X_0 X_2 X_4 X_5 X_6 X_8 X_9 X_{12}
R_3 R_8
X_1 X_7
[DRS Graph Representation]
SPLIT
ROOT ROOT
DELETION
ROOT ROOT
X_2' X_9'
In 1964 Peter Higgs published his paper describing Higgs mechanism
Higgs mechanism predicted a new boson
PBMT+LM
Peter Higgs wrote his paper explaining Higgs mechanism in 1964 .
Higgs mechanism predicted a new elementary particle .

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/a
X_3	5	publish/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, spin-zero/a massive/a, boson/n
X_{10}	17	predict/v, **event**
X_{11}	13	describe/v, **event**
X_{12}	$24,25,26$	first/a, time/n
O_1	16	which/WDT

rel	pos. in S	predicate
R_1	5	$agent, X_3 \to X_0$
R_2	5	$patient, X_3 \to X_2$
R_3	6	$of, X_2 \to X_1$
R_4	9	$in, X_3 \to X_4$
R_5	1	$in, X_3 \to X_5$
R_6	13	$agent, X_{11} \to X_6$
R_7	13	$patient, X_{11} \to X_8$
R_8	$--$	$nn, X_8 \to X_7$
R_9	17	$agent, X_{10} \to X_8$
R_{10}	17	$patient, X_{10} \to X_9$
R_{11}	23	$for, X_{10} \to X_{12}$

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 ."*

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 in the graph tables). Orphan words (OW) 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 matches the positions in the input sentence and thus deriving the input sentence.

Split Candidate	isSplit	prob.
$(agent, for, patient)$ - $(agent, in, in, patient)$	true	0.63
	false	0.37

Table 1: Simplification: SPLIT

Given the input DRS shown in Figure 1, simplification proceeds as follows.

Splitting. The splitting candidates of a DRS are event pairs contained in that DRS. More precisely, the splitting candidates are pairs[4] of event variables associated with at least one of the core thematic roles (e.g., *agent* and *patient*). The features conditioning a split are the set of thematic roles associated with each event variable. The DRS shown in Figure 1 contains three such event variables X_3, X_{11} and X_{10} with associated thematic role sets {*agent, in, in, patient*}, {*agent, patient*} and {*agent, for, patient*} respectively. Hence, there are 3 splitting candidates (X_3-X_{11}, X_3-X_{10} and X_{10}-X_{11}) and 4 split options: no split or split at one of the splitting candidates. Here the split with highest probability (cf. Table 1) is chosen and the DRS is split into two sub-DRS, one containing X_3, and the other containing X_{10}. After splitting, dangling subgraphs are attached to the root of the new subgraph maximizing either proximity or position overlap. Here the graph rooted in X_{11} is attached to the root dominating X_3 and the orphan word O_1 to the root dominating X_{10}.

Deletion. The deletion model (cf. Table 2) regulates the deletion of relations and their associated subgraph; of adjectives and adverbs; and of orphan words. Here, the relations *in* between X_3 and X_4 and*for* between X_{10} and X_{12} are deleted resulting in the deletion of the phrases *"in Physical Review Letters"* and *"for the first time"* as well as the adjectives *second, massive, spin-zero* and the orphan word *which.*

Substitution and Reordering. Finally the translation and language model ensures that *published, describing* and *boson* are simplified to *wrote, explaining* and *elementary particle* respectively; and that the phrase *"In 1964"* is moved from the beginning of the sentence to its end.

3.2 The Simplification Model

Our simplification framework consists of a probabilistic model for splitting and dropping which we call DRS simplification model (DRS-SM); a phrase based translation model for substitution and reordering (PBMT); and a language model learned on Simple English Wikipedia (LM) for fluency and grammaticality. Given a complex sentence c, we split the simplification process into two steps. First, DRS-SM is applied to D_c (the DRS representation of the complex sentence c) to produce one or more (in case of splitting) intermediate simplified sentence(s) s'. Second, the simplified sentence(s) s' is further simplified to s using a phrase based machine translation system (PBMT+LM). Hence, our model can be formally defined as:

$$\begin{aligned}\hat{s} &= \operatorname*{argmax}_{s} p(s|c)\\ &= \operatorname*{argmax}_{s} \sum_{s'} p(s'|c)p(s|s')\\ &= \operatorname*{argmax}_{s} \sum_{s'} p(s'|\mathrm{D}_c)p(s'|s)p(s)\end{aligned}$$

where the probabilities $p(s'|\mathrm{D}_c)$, $p(s'|s)$ and $p(s)$ are given by the DRS simplification model, the phrase based machine translation model and the language model respectively.

To get the DRS simplification model, we combine the probability of splitting with the probability of deletion:

$$p(s'|\mathrm{D}_c) = \sum_{\theta:str(\theta(\mathrm{D}_c))=s'} p(\mathrm{D}_{split}|\mathrm{D}_c)p(\mathrm{D}_{del}|\mathrm{D}_{split})$$

where θ is a sequence of simplification operations and $str(\theta(\mathrm{D}_c))$ is the sequence of words associated with a DRS resulting from simplifying D_c using θ.

The probability of a splitting operation for a given DRS D_c is:

$$p(\mathrm{D}_{split}|\mathrm{D}_c) = \begin{cases} \mathrm{SPLIT}(sp^{\text{true}}_{\text{cand}}), & \text{split at } sp_{\text{cand}}\\ \prod_{sp_{\text{cand}}} \mathrm{SPLIT}(sp^{\text{false}}_{\text{cand}}), & \text{otherwise}\end{cases}$$

[4] The splitting candidates could be sets of event variables depending on the number of splits required. Here, we consider pairs for 2 splits.

relation candidate		isDrop	prob.
relation word	length range		
in	0-2	true	0.22
		false	0.72
in	2-5	true	0.833
		false	0.167

mod. cand.	isDrop	prob.
mod word		
new	true	0.22
	false	0.72
massive	true	0.833
	false	0.167

OW candidate		isDrop	prob.
orphan word	isBoundary		
and	true	true	0.82
		false	0.18
which	false	true	0.833
		false	0.167

Table 2: Simplification: DELETION (Relations, modifiers and OW respectively)

That is, if the DRS is split on the splitting candidate sp_{cand}, the probability of the split is then given by the *SPLIT* table (Table 1) for the *isSplit* value "true" and the split candidate sp_{cand}; else it is the product of the probability given by the *SPLIT* table for the *isSplit* value "false" for all split candidate considered for D_c. As mentioned above, the features used for determining the split operation are the role sets associated with pairs of event variables (cf. Table 1).

The deletion probability is given by three models: a model for relations determining the deletion of prepositional phrases; a model for modifiers (adjectives and adverbs) and a model for orphan words (Table 2). All three deletion models use the associated word itself as a feature. In addition, the model for relations uses the PP length-range as a feature while the model for orphan words relies on boundary information i.e., whether or not, the OW occurs at the associated sentence boundary.

$$p(\mathrm{D}_{del}|\mathrm{D}_{split}) = \prod_{\mathrm{rel_{cand}}} \mathrm{DEL_{rel}}(\mathrm{rel_{cand}}) \prod_{\mathrm{mod_{cand}}} \mathrm{DEL_{mod}}(\mathrm{mod_{cand}}) \prod_{\mathrm{ow_{cand}}} \mathrm{DEL_{ow}}(\mathrm{ow_{cand}})$$

3.3 Estimating the parameters

We use the EM algorithm (Dempster et al., 1977) to estimate our split and deletion model parameters. For an efficient implementation of EM algorithm, we follow the work of Yamada and Knight (2001) and Zhu et al. (2010); and build training graphs (Figure 2) from the pair of complex and simple sentence pairs in the training data.

Each training graph represents a complex-simple sentence pair and consists of two types of nodes: major nodes (M-nodes) and operation nodes (O-nodes). An M-node contains the DRS representation D_c of a complex sentence c and the associated simple sentence(s) s_i while O-nodes determine split and deletion operations on their parent M-node. Only the root M-node is considered for the split operations. For example, given the root M-node $(\mathrm{D}_c, (s_1, s_2))$, multiple successful split O-nodes will be created, each one further creating two M-nodes (D_{c1}, s_1) and (D_{c2}, s_2). For the training pair (c, s), the root M-node (D_c, s) is followed by a single split O-node producing an M-node (D_c, s) and counting all split candidates in D_c for failed split. The M-nodes created after split operations are then tried for multiple deletion operations of relations, modifiers and OW respectively. Each deletion candidate creates a deletion O-node marking successful or failed deletion of the candidate and a result M-node. The deletion process continues on the result M-node until there is no deletion candidate left to process. The governing criteria for the construction of the training graph is that, at each step, it tries to minimize the Levenshtein edit distance between the complex and the simple sentences. Moreover, for the splitting operation, we introduce a split only if the reference sentence consists of several sentences (i.e., there is a split in the training data); and only consider splits which maximises the overlap between split and simple reference sentences.

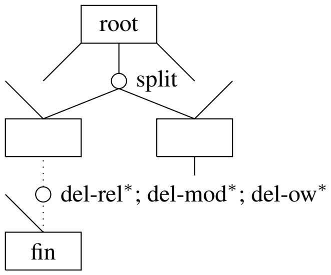


Figure 2: An example training graph

We initialize our probability tables Table 1 and Table 2 with the uniform distribution, i.e., 0.5 because all our features are binary. The EM algorithm iterates over training graphs counting model features from O-nodes and updating our probability tables. Because of the space constraints, we do not describe our algorithm in details. We refer the reader to (Yamada and Knight, 2001) for more details.

Our phrase based translation model is trained using the Moses toolkit[5] with its default command line options on the PWKP corpus (except the sentences from the test set) considering the complex sentence as the source and the simpler one as the target. Our trigram language model is trained using the SRILM toolkit[6] on the SWKP corpus[7].

Decoding. We explore the decoding graph similar to the training graph but in a greedy approach always picking the choice with maximal probability. Given a complex input sentence c, a split O-node will be selected corresponding to the decision of whether to split and where to split. Next, deletion O-nodes are selected indicating whether or not to drop each of the deletion candidate. The DRS associated with the final M-node D_{fin} is then mapped to a simplified sentence s'_{fin} which is further simplified using the phrase-based machine translation system to produce the final simplified sentence s_{simple}.

4 Experiments

We trained our simplification and translation models on the PWKP corpus. To evaluate performance, we compare our approach with three other state of the art systems using the test set provided by Zhu et al. (2010) and relying both on automatic metrics and on human judgments.

4.1 Training and Test Data

The DRS-Based simplification model is trained on PWKP, a bi-text of complex and simple sentences provided by Zhu et al. (2010). To construct this bi-text, Zhu et al. (2010) extracted complex and simple sentences from EWKP and SWKP respectively and automatically aligned them using TF*IDF as a similarity measure. PWKP contains $108016/114924$ complex/simple sentence pairs. We tokenize PWKP using Stanford CoreNLP toolkit[8]. We then parse all complex sentences in PWKP using Boxer[9] to produce their DRSs. Finally, our DRS-Based simplification model is trained on 97.75% of PWKP; we drop out 2.25% of the complex sentences in PWKP which are repeated in the test set or for which Boxer fails to produce DRSs.

We evaluate our model on the test set used by Zhu et al. (2010) namely, an aligned corpus of $100/131$ EWKP/SWKP sentences. Boxer produces a DRS for 96 of the 100 input sentences. These input are simplified using our simplification system namely, the DRS-SM model and the phrase-based machine translation system (Section 3.2). For the remaining four complex sentences, Boxer fails to produce DRSs. These four sentences are directly sent to the phrase-based machine translation system to produce simplified sentences.

4.2 Automatic Evaluation Metrics

To assess and compare simplification systems, two main automatic metrics have been used in previous work namely, BLEU and the Flesch-Kincaid Grade Level Index (FKG).

The FKG index is a readability metric taking into account the average sentence length in words and the average word length in syllables. In its original context (language learning), it was applied to well formed text and thus measured the simplicity of a well formed sentence. In the context of the simplification task however, the automatically generated sentences are not necessarily well formed so that the FKG index reduces to a measure of the sentence length (in terms of words and syllables) approximating the simplicity level of an output sentence irrespective of the length of the corresponding input. To assess simplification, we instead use metrics that are directly related to the simplification task namely, the number of splits in the overall (test and training) data and in average per sentences; the number of generated sentences with no edits i.e., which are identical to the original, complex one; and the average Levenshtein distance between the system's output and both the complex and the simple reference sentences.

BLEU gives a measure of how close a system's output is to the gold standard simple sentence. Because there are many possible ways of simplifying a sentence, BLEU alone fails to correctly assess the appropriateness of a simplification. Moreover BLEU does not capture the degree to which the system's output differs from the complex sentence input. We therefore use BLEU as a means to evaluate how close the systems output are to the reference corpus but complement it with further manual metrics capturing other important factors when

[5] http://www.statmt.org/moses/

[6] http://www.speech.sri.com/projects/srilm/

[7] We downloaded the snapshots of Simple Wikipedia dated 2013-10-30 available at http://dumps.wikimedia.org/.

[8] http://nlp.stanford.edu/software/corenlp.shtml

[9] http://svn.ask.it.usyd.edu.au/trac/candc, Version 1.00

evaluating simplifications such as the fluency and the adequacy of the output sentences and the degree to which the output sentence simplifies the input.

4.3 Results and Discussion

Number of Splits Table 3 shows the proportion of input whose simplification involved a splitting operation. While our system splits in proportion similar to that observed in the training data, the other systems either split very often (80% of the time for Zhu and 63% of the time for Woodsend) or not at all (Wubben). In other words, when compared to the other systems, our system performs splits in proportion closest to the reference both in terms of total number of splits and of average number of splits per sentence.

Data	Total number of sentences	% split	average split / sentence
PWKP	108,016	6.1	1.06
GOLD	100	28	1.30
Zhu	100	80	1.80
Woodsend	100	63	2.05
Wubben	100	1	1.01
Hybrid	100	10	1.10

Table 3: Proportion of Split Sentences (% split) in the training/test data and in average per sentence (average split / sentence). GOLD is the test data with the gold standard SWKP sentences; Zhu, Woodsend, Wubben are the best output of the models of Zhu et al. (2010), Woodsend and Lapata (2011) and Wubben et al. (2012) respectively; Hybrid is our model.

Number of Edits Table 4 indicates the edit distance of the output sentences w.r.t. both the complex and the simple reference sentences as well as the number of input for which no simplification occur. The right part of the table shows that our system generate simplifications which are closest to the reference sentence (in terms of edits) compared to those output by the other systems. It also produces the highest number of simplifications which are identical to the reference. Conversely our system only ranks third in terms of dissimilarity with the input complex sentences (6.32 edits away from the input sentence) behind the Woodsend (8.63 edits) and the Zhu (7.87 edits) system. This is in part due to the difference in splitting strategies noted above : the many splits applied by these latter two systems correlate with a high number of edits.

System	BLEU	Edits (Complex to System)		Edits (System to Simple)	
		LD	No edit	LD	No edit
GOLD	100	12.24	3	0	100
Zhu	37.4	7.87	2	14.64	0
Woodsend	42	8.63	24	16.03	2
Wubben	41.4	3.33	6	13.57	2
Hybrid	53.6	6.32	4	11.53	3

Table 4: Automated Metrics for Simplification: average Levenshtein distance (LD) to complex and simple reference sentences per system ; number of input sentences for which no simplification occur (No edit).

BLEU score We used Moses support tools: multi-bleu[10] to calculate BLEU scores. The BLEU scores shown in Table 4 show that our system produces simplifications that are closest to the reference.

In sum, the automatic metrics indicate that our system produces simplification that are consistently closest to the reference in terms of edit distance, number of splits and BLEU score.

4.4 Human Evaluation

The human evaluation was done online using the LG-Eval toolkit (Kow and Belz, 2012)[11]. The evaluators were allocated a trial set using a Latin Square Experimental Design (LSED) such that each evaluator sees the same number of output from each system and for each test set item. During the experiment, the evaluators were presented with a pair of a complex and a simple sentence(s) and asked to rate this pair w.r.t. to adequacy (Does the simplified sentence(s) preserve the meaning of the input?) and simplification (Does the generated sentence(s) simplify the complex input?). They were also asked to rate the second (simplified) sentence(s) of the pair w.r.t. to fluency (Is the simplified output fluent and grammatical?). Similar to the Wubben's human evaluation setup, we randomly selected 20 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 (Hybrid) and the output of the other three systems (Zhu, Woodsend and Wubben) which were provided to us by the system authors. The evaluation data thus consisted of 100 complex/simple pairs. We collected ratings from 27 participants.

[10] http://www.statmt.org/moses/?n=Moses.SupportTools

[11] http://www.nltg.brighton.ac.uk/research/lg-eval/

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.

Systems	Simplification	Fluency	Adequacy
GOLD	3.57	3.93	3.66
Zhu	2.84	2.34	2.34
Woodsend	1.73	2.94	3.04
Wubben	1.81	3.65	3.84
Hybrid	3.37	3.55	3.50

Table 5: Average Human Ratings for simplicity, fluency and adequacy

Table 5 shows the average ratings of the human evaluation on a slider 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 and are shown in Table 6.

Systems	GOLD	Zhu	Woodsend	Wubben
Zhu	♢□△			
Woodsend	♢□△	♢□△		
Wubben	♢■▲	♢□△	♦□△	
Hybrid	♦■▲	♦□△	♢□▲	♢■▲

Table 6: ♢/♦ is/not significantly different (sig. diff.) wrt simplicity. □/■ is/not sig. diff. wrt fluency. △/▲ is/not sig. diff. wrt adequacy. (significance level: $p < 0.05$)

With regard to simplification, our system ranks first and is very close to the manually simplified input (the difference is not statistically significant). The low rating for Woodsend reflects the high number of unsimplified sentences (24/100 in the test data used for the automatic evaluation and 6/20 in the evaluation data used for human judgments). Our system data is not significantly different from the manually simplified data for simplicity whereas all other systems are.

For fluency, our system rates second behind Wubben and before Woodsend and Zhu. The difference between our system and both Zhu and Woodsend system is significant. In particular, Zhu's output is judged less fluent probably because of the many incorrect splits it licenses. Manual examination of the data shows that Woodsend's system also produces incorrect splits. For this system however, the high proportion of non simplified sentences probably counterbalances these incorrect splits, allowing for a good fluency score overall.

Regarding adequacy, our system is against closest to the reference (3.50 for our system vs. 3.66 for manual simplification). Our system, the Wubben system and the manual simplifications are in the same group (the differences between these systems are not significant). The Woodsend system comes second and the Zhu system third (the difference between the two is significant). Wubben's high fluency, high adequacy but low simplicity could be explained with their minimal number of edit (3.33 edits) from the source sentence.

In sum, if we group together systems for which there is no significant difference, our system ranks first (together with GOLD) for simplicity; first for fluency (together with GOLD and Wubben); and first for adequacy (together with GOLD and Wubben).

5 Conclusion

A key feature of our approach is that it is semantically based. Typically, discourse level simplification operations such as sentence splitting, sentence reordering, cue word selection, referring expression generation and determiner choice are semantically constrained. As argued by Siddharthan (2006), correctly capturing the interactions between these phenomena is essential to ensuring text cohesion. In the future, we would like to investigate how our framework deals with such discourse level simplifications i.e., simplifications which involves manipulation of the coreference and of the discourse structure. In the PWKP data, the proportion of split sentences is rather low (6.1 %) and many of the split sentences are simple sentence coordination splits. A more adequate but small corpus is that used in (Siddharthan, 2006) which consists of 95 cases of discourse simplification. Using data from the language learning or the children reading community, it would be interesting to first construct a similar, larger scale corpus; and to then train and test our approach on more complex cases of sentence splitting.

Acknowledgments

We are grateful to Zhemin Zhu, Kristian Woodsend and Sander Wubben for sharing their data. We would like to thank our annotators for participating in our human evaluation experiments and to anonymous reviewers for their insightful comments.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 597–604. Association for Computational Linguistics.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 25–32. Association for Computational Linguistics.

Stefan Bott, Horacio Saggion, 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. Citeseer.

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.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

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.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, page 896. 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. 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)*, 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 Mattos 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.