Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming

Kristian Woodsend and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
k.woodsend@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

Text simplification aims to rewrite text into simpler versions, and thus make information accessible to a broader audience. Most previous work simplifies sentences using hand-crafted rules aimed at splitting long sentences, or substitutes difficult words using a predefined dictionary. This paper presents a data-driven model based on quasi-synchronous grammar, a formalism that can naturally capture structural mismatches and complex rewrite operations. We describe how such a grammar can be induced from Wikipedia and propose an integer linear programming model for selecting the most appropriate simplification from the space of possible rewrites generated by the grammar. We show experimentally that our method creates simplifications that significantly reduce the reading difficulty of the input, while maintaining grammaticality and preserving its meaning.

1 Introduction

Sentence simplification is perhaps one of the oldest text rewriting problems. Given a *source* sentence, the goal is to create a grammatical *target* that is easier to read with simpler vocabulary and syntactic structure. An example is shown in Table 1 involving a broad spectrum of rewrite operations such as deletion, substitution, insertion, and reordering. The popularity of the simplification task stems from its potential relevance to various applications. Examples include the development of reading aids for people with aphasia (Carroll et al., 1999), non-native speakers (Siddharthan, 2003) and more generally individuals with low literacy (Watanabe et al., 2009). A simplification component could be also used as a preprocessing step to improve the performance of parsers (Chandrasekar et al., 1996), summarizers (Beigman Klebanov et al., 2004) and semantic role labelers (Vickrey and Koller, 2008).

Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents, which precedes the full purchasing agents report that is due out today and gives an indication of what the full report might hold.
Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. The Chicago report precedes the full purchasing agents report. The Chicago report gives an indication of what the full report might hold. The full report is due out today.

Table 1: Example of a source sentence (top) and its simplification (bottom).

Simplification is related to, but different from paraphrase extraction (Barzilay, 2003). We must not only have access to paraphrases (i.e., rewrite rules), but also be able to combine them to generate new text, in a simpler language. The task is also distinct from sentence compression as it aims to render a sentence more accessible while preserving its meaning. On the contrary, compression unavoidably leads to some information loss as it creates shorter sentences without necessarily reducing complexity. In fact, one of the commonest simplification operations is sentence splitting which usually produces longer rather than shorter output! Moreover, mod-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 409–420,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

els developed for sentence compression have been mostly designed with one rewrite operation in mind, namely word deletion, and are thus unable to model consistent syntactic effects such as reordering, sentence splitting, changes in non-terminal categories, and lexical substitution (but see Cohn and Lapata 2008 and Zhao et al. 2009 for notable exceptions).

In this paper we propose a sentence simplification model that is able to handle structural mismatches and complex rewriting operations. Our approach is based on quasi-synchronous grammar (QG, Smith and Eisner 2006), a formalism that is well suited for text rewriting. Rather than postulating a strictly synchronous structure over the source and target sentences, QG identifies a "sloppy" alignment of parse trees assuming that the target tree is in some way "inspired by" the source tree. Specifically, our model is formulated as an integer linear program and uses QG to capture the space of all possible rewrites. Given a source tree, it finds the best target tree licensed by the grammar subject to constraints such as sentence length and reading ease. Our model is conceptually simple and computationally efficient. Furthermore, it finds globally optimal simplifications without resorting to heuristics or approximations during the decoding process.

Contrary to most previous approaches (see the discussion in Section 2) which rely heavily on hand-crafted rules, our model *learns* simplification rewrites automatically from examples of source-target sentences. Our work joins others in using Wikipedia to extract data appropriate for model training (Yamangil and Nelken, 2008; Yatskar et al., 2010; Zhu et al., 2010). Advantageously, the Simple English Wikipedia (henceforth SimpleEW) provides a large repository of simplified language; it uses fewer words and simpler grammar than the ordinary English Wikipedia (henceforth MainEW) and is aimed at non-native English speakers, children, translators, people with learning disabilities or low reading proficiency. We exploit Wikipedia and create a (parallel) simplification corpus in two ways: by aligning MainEW sentences to their SimpleEW counterparts, and by extracting training instances from SimpleEW revision histories, thus leveraging Wikipedia's collaborative editing process.

Our experimental results demonstrate that a simplification model can be learned from Wikipedia data alone without any manual effort. Perhaps unsurprisingly, the quality of the QG grammar rules greatly improves when these are learned from revision histories which are less noisy than sentence alignments. When compared against current state-of-the-art methods (Zhu et al., 2010) our model yields significantly simpler output that is both grammatical and meaning preserving.

2 Related Work

Sentence simplification has attracted a great deal of attention due to its potential impact on society. The literature is rife with attempts to simplify text using mostly hand-crafted syntactic rules aimed at splitting long and complicated sentences into several simpler ones (Carroll et al., 1999; Chandrasekar et al., 1996; Siddharthan, 2004; Vickrey and Koller, 2008). Other work focuses on lexical simplifications and substitutes difficult words by more common WordNet synonyms or paraphrases found in a predefined dictionary (Devlin, 1999; Inui et al., 2003; Kaji et al., 2002).

More recently, Yatskar et al. (2010) explore data-driven methods to learn lexical simplifications from Wikipedia revision histories. A key idea in their work is to utilize SimpleEW edits, while recognizing that these may serve other functions, such as vandalism removal or introduction of new content. Zhu et al. (2010) also use Wikipedia to learn a sentence simplification model which is able to perform four rewrite operations, namely substitution, reordering, splitting, and deletion. Inspired by syntax-based SMT (Yamada and Knight, 2001), their model consists of three components: a language model $P(\mathbf{s})$ whose role is to guarantee that the simplification output is grammatical, a direct translation model $P(\mathbf{s}|\mathbf{c})$ capturing the probability that the target sentence $\mathbf{s}$ is a simpler version of the source $\mathbf{c}$, and a decoder which searches for the simplification $\mathbf{s}$ which maximizes $P(\mathbf{s})P(\mathbf{s}|\mathbf{c})$. The translation model is the product of the aforementioned four rewrite operations whose probabilities are estimated from a parallel corpus of MainEW and SimpleEW sentences using an expectation maximization algorithm. Their decoder translates sentences into simpler alternatives by greedily selecting the branch in the source tree with the highest probability.

Our own work formulates sentence simplification in the framework of Quasi-synchronous grammar (QG, Smith and Eisner 2006). QG allows to describe non-isomorphic tree pairs (the grammar rules can comprise trees of arbitrary depth, and fragments can be mapped) and is thus suited to text-rewriting tasks which typically involve a number of local modifications to the input text. We use quasi-synchronous grammar to learn a wide range of rewrite operations capturing both lexical and structural simplifications naturally without any additional rule engineering. In contrast to Yatskar et al. (2010) and Zhu et al. (2010), simplification operations (e.g., substitution or splitting) are not modeled explicitly; instead, we leave it up to our grammar extraction algorithm to learn appropriate rules that reflect the training data. Compared to Zhu et al., our model is conceptually simpler and more general. The proposed ILP formulation not only allows to efficiently search through the space of many QG rules but also to incorporate constraints relating to grammaticality and the task at hand without the added computational cost of integrating a language model. Furthermore, our learning framework is not limited to simplification and could be easily adapted to other rewriting tasks. Indeed, the QG formalism has been previously applied to parser adaptation and projection (Smith and Eisner, 2009), paraphrase identification (Das and Smith, 2009), question answering (Wang et al., 2007), and title generation (Woodsend et al., 2010).

Finally, our work relates to a large body of recent literature on Wikipedia and its potential for a wide range of NLP tasks. Beyond text rewriting, examples include semantic relatedness (Ponzetto and Strube, 2007), information extraction (Wu and Weld, 2010), ontology induction (Nastase and Strube, 2008), and the automatic creation of overview articles (Sauper and Barzilay, 2009).

3 Sentence Simplification Model

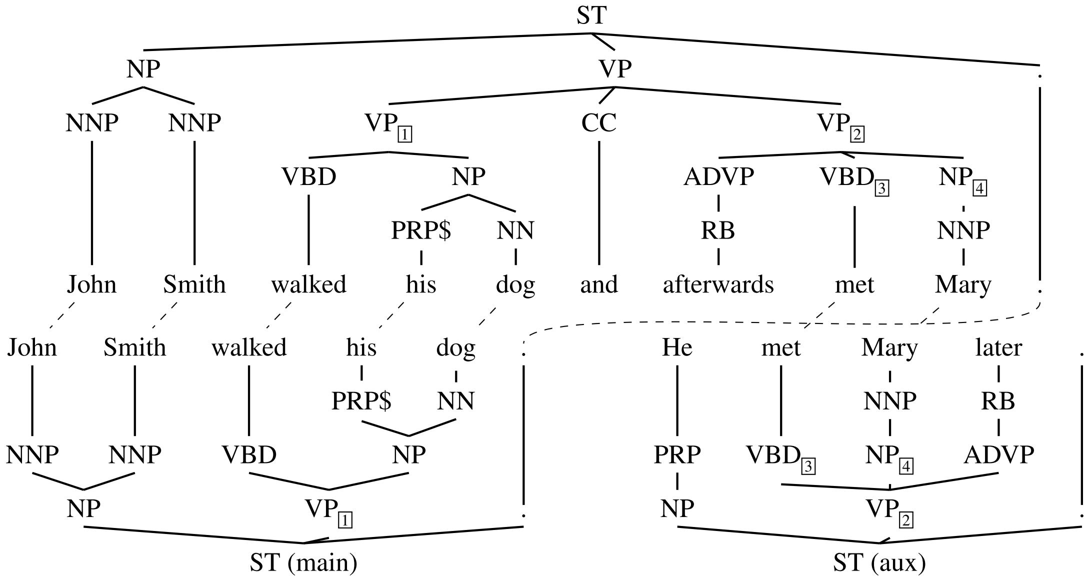
Our model takes a single sentence as input and creates a version that is simpler to read. This may involve rendering syntactically complex structures simpler (e.g., through sentence splitting), or substituting rare words with more common words or phrases (e.g., such that a second language learner may be familiar with), or deleting elements of the original text in order to produce a relatively simpler and shallower syntactic structure. In addition, the output must be grammatical and coherent. These constraints are *global* in their scope, and cannot be adequately satisfied by optimizing each one of them individually. Our approach therefore uses an ILP formulation which will provide a globally optimal solution. Given an input sentence, our model deconstructs it into component phrases and clauses, each of which is simplified (lexically and structurally) through QG rewrite rules. We generate all possible simplifications for a given input and use the ILP to find the best target subject to grammaticality constraints. In what follows we first detail how we extract QG rewrite rules as these form the backbone of our model and then formulate the ILP proper.

3.1 Quasi-synchronous Grammar

Phrase alignment Our model operates on individual sentences annotated with syntactic information i.e., phrase structure trees. In our experiments, we obtain this information from the Stanford parser (Klein and Manning, 2003) but any other broadly similar parser could be used instead. Given an input sentence S1 or its parse tree T1, the QG constructs a monolingual grammar for parsing, or generating, possible translation trees T2. A grammar node in the target tree T2 is modeled on a subset of nodes in the source tree, with a rather loose alignment between the trees.

We take aligned sentence pairs represented as phrase structure trees and build up a list of leaf node alignments based on lexical identity. We align direct parent nodes where more than one child node aligns. QG rules are created from aligned nodes above the leaf node level if the all the nodes in the target tree can be explained using nodes from the source. This helps to improve the quality in what is inherently a noisy process, and it is largely responsible for a relatively small resulting grammar (see Table 2). Examples of phrase alignments (indicated with dotted lines) are shown in Figure 1.

Syntactic simplification rules Each QG rule describes the transformations required from source to target phrase sub-trees. It allows child (and possibly grand-child) constituents to be deleted or re-



Rule involving lexical substitution:
⟨VP, VP⟩ → ⟨[ADVP [RB *afterwards*] VBD$_{\boxed{3}}$ NP$_{\boxed{4}}$], [VBD$_{\boxed{3}}$ NP$_{\boxed{4}}$ ADVP [RB *later*]]⟩

Rule for splitting into main constituent and auxiliary sentence:
⟨VP, VP, ST⟩ → ⟨[VP$_{\boxed{1}}$ *and* VP$_{\boxed{2}}$], [VP$_{\boxed{1}}$], [NP [PRP *He*] VP$_{\boxed{2}}$.]⟩

Figure 1: A source sentence (upper tree) is split into two sentences. Dotted lines show word alignments, while boxed subscripts show aligned nodes used to form QG rules. Below, two QG rules learned from this data.

ordered, and for nodes to be flattened. In addition, we allow insertion of punctuation and some function words, identified by a small set of POS tags. To distinguish sentences proper (which have final punctuation) from clauses, we modify the output of the parser, changing the root sentence parse tag from S to ST (a "top-level sentence"); this allows clauses to be extracted and rewritten as stand-alone sentences.

Lexical simplification rules Lexical substitutions are an important part of simplification. We learn them from aligned sub-trees, in the same way as described above for syntax rules, by allowing a small number of lexical substitutions to be present in the rules, and provided they do not include proper nouns. The resulting QG rules could be applied by matching the syntax of the whole sub-tree surrounding the substitution, but this approach is overly restrictive and suffers from data sparsity. Indeed, Yatskar et al. (2010) learn lexical simplifications without taking syntactic context into account. We therefore add a post-processing stage to the learning process. For rules where the syntactic structures of the source and target sub-trees match, and the only difference is a lexical substitution, we construct a more general rule by extracting the words and corresponding POS tags involved in the substitution. Then at the generation stage, identifying suitable rules depends only on the substitution words, rather than the surrounding syntactic context. An example of a lexical substitution rule is shown in Figure 1.

Sentence splitting rules Another important simplification technique is to split syntactically complicated sentences into several shorter ones. To learn QG rules for this operation, the source sentence is aligned with two consecutive target sentences.

Rather than expecting to discover a split point in the source sentence, we attempt to identify a node in the source parse tree that contributes to both of the two target sentences. Our intuition is that one of the target sentences will follow the general syntactic structure of the source sentence. We designate this as the *main* sentence. A node in the source sentence parse tree will be aligned with a (similar but simpler) node in the main target sentence, but at the same time it will fully explain the other target sentence, which we term the *auxiliary* sentence. It is possible for the auxiliary sentence to come before or after the main sentence. In the learning procedure, we try both possible orderings, and record the order in any QG rules successfully produced.

The resulting QG rule is a tuple of three phrase structure elements: the source node, the node in the target main sentence (the top level of this node is typically the same as that of the source node), and the phrase structure of the entire auxiliary sentence.[1] In addition, there is a flag to indicate if the auxiliary sentence comes before or after the main sentence. This formalism is able to capture the operations required to split sentences containing coordinate or subordinate clauses, parenthetical content, relative clauses and apposition. An example of a sentence splitting rule is illustrated in Figure 1.

3.2 ILP-based Generation

We cast the problem of finding a suitable target simplification given a source sentence as an integer linear program (ILP). Specifically, simplified text is created from source sentence parse trees by identifying and applying QG grammar rules. These will have matching structure and may also require lexical matching (shown using italics in the example rules in Figure 1). The generation process starts at the root node of the parse tree, applying QG rules to subtrees until leaf nodes are reached. We do not use the Bayesian probability model proposed by Smith and Eisner (2006) to identify the best sequence of simplification rules. Instead, where there is more than one matching rule, and so more than one simplification is possible, the alternatives are all generated and incorporated into the target phrase structure tree. The ILP model operates over this phrase structure tree and selects the phrase nodes from which to form the target output.

Applying the QG rules on the source sentence generates a number of auxiliary sentences. Let $\mathcal{S}$ be this set of sentences. Let $\mathcal{P}$ be the set of nodes in the phrase structure trees of the auxiliary sentences, and $\mathcal{P}_s \subset \mathcal{P}$ be the set of nodes in each sentence $s \in \mathcal{S}$. Let the sets $\mathcal{D}_i \subset \mathcal{P}, \forall i \in \mathcal{P}$ capture the phrase dependency information for each node i, where each set $\mathcal{D}_i$ contains the nodes that depend on the presence of i. In a similar fashion, the sets $\mathcal{A}_i \subset \mathcal{S}, \forall i \in \mathcal{P}$ capture the indices of any auxiliary sentences that depend on the presence of node i. $\mathcal{C} \subset \mathcal{P}$ is the set of nodes involving a choice of alternative simplifications (nodes in the tree where more than one QG rewrite rule can be applied, as mentioned above); $\mathcal{C}_i \subset \mathcal{P}, i \in \mathcal{C}$ are the sets of nodes that are direct children of each such node, in other words they are the individual simplifications. Let $l_i^{(w)}$ be the length of each node i in words, and $l_i^{(sy)}$ its length in syllables. As we shall see below counts of words and syllables are important cues in assessing readability.

The model is cast as an binary integer linear program. A vector of binary decision variables $x \in \{0,1\}^{|\mathcal{P}|}$ indicates if each node is to be part of the output. A vector of auxiliary binary variables $y \in \{0,1\}^{|\mathcal{S}|}$ indicates which (auxiliary) sentences have been chosen.

$$\begin{aligned}
\max_x \quad & \sum_{i \in \mathcal{P}} g_i x_i + h_w + h_{sy} && && \text{(1a)}\\
\text{s.t.} \quad & x_j \rightarrow x_i && \forall i \in \mathcal{P}, j \in \mathcal{D}_i && \text{(1b)}\\
& x_i \rightarrow y_s && \forall i \in \mathcal{P}, s \in \mathcal{A}_i && \text{(1c)}\\
& x_i \rightarrow y_s && \forall s \in \mathcal{S}, i \in \mathcal{P}_s && \text{(1d)}\\
& \sum_{j \in \mathcal{C}_i} x_j = x_i && \forall i \in \mathcal{C}, j \in \mathcal{C}_i && \text{(1e)}\\
& \sum_{s \in \mathcal{S}} y_i \geq 1 && && \text{(1f)}\\
& x_i \in \{0,1\} && \forall i \in \mathcal{P} && \text{(1g)}\\
& y_s \in \{0,1\} && \forall s \in \mathcal{S}. && \text{(1h)}
\end{aligned}$$

Our objective function, given in Equation (1a), is the summation of local and global components. Each phrase is locally given a *rewrite penalty* g_i, where common lexical substitutions, rewrites and simplifications are penalized less (as we trust them more), compared to rarer QG rules. The penalty is a simple log-probability measure, $g_i = \log\left(\frac{n_r}{N_r}\right)$, where n_r is the number of times the QG rule r was seen in the training data, and N_r the number of times all suitable rules for this phrase node were seen. If no suitable rules exist, we set $g_i = 0$.

The other two components of the objective, h_w and h_{sy}, are global in nature, and guide the ILP

[1] Note that the target component comprises the second and third elements as a pair, and variables from the source component are split between them.

towards simpler language. They draw inspiration from existing measures of readability (the ease with which a document can be read and understood). The primary aim of readability formulas is to assess whether texts or books are suitable for students at particular grade levels or ages (see Mitchell 1985 for an overview). Intuitively, texts ought to be simpler if they correspond to low reading levels. A commonly used reading level measure is the Flesch-Kincaid Grade Level (FKGL) index which estimates readability as a combination of the average number of syllables per word and the average number of words per sentence. Unfortunately, this measure is non-linear[2] and cannot be incorporated directly into the objective of the ILP. Instead, we propose a linear approximation. We provide the ILP with targets for the average number of words per sentence (wps), and syllables per word (spw). $h_w(x,y)$ then measures the number of words below this target level that the ILP has achieved:

$$h_w(x,y) = \text{wps} \times \sum_{i \in \mathcal{S}} y_i - \sum_{i \in \mathcal{P}} l_i^{(w)} x_i.$$

When positive, this indicates that sentences are shorter than target, and contributes positively to the readability objective whilst encouraging the application of sentence splitting and deletion-based QG rules. Similarly, $h_{sy}(x,y)$ measures the number of syllables below that expected, from the target average and the number of words the ILP has chosen:

$$h_{sy}(x) = \text{spw} \times \sum_{i \in \mathcal{P}} l_i^{(w)} x_i - \sum_{i \in \mathcal{P}} l_i^{(sy)} x_i.$$

This component of the objective encourages the deletion or lexical substitution of complex words. We can use the two target parameters (wps and spw) to control how much simplification the ILP should apply.

Constraint (1b) enforces grammatical correctness by ensuring that the phrase dependencies are respected and the resulting structure is a tree. Phrases that depend on phrase i are contained in the set $\mathcal{D}_i$. Variable x_i is true, and therefore phrase i will be included in the target output, if any of its dependents $x_j \in \mathcal{D}_i$ are true.[3] Constraint (1c) links main phrases to auxiliary sentences, so that the latter can only be included in the output if the main phrase has also been chosen. This helps to control coherence within the output text. Despite seeming similar to (1c), the role of constraint (1d) is quite different. It links phrase variables x to sentence variables y, to ensure the logical integrity of the model is correct. Where the QG provides alternative simplifications, it makes sense of course to select only one. This is controlled by constraint (1e), and by placing all alternatives in the set $\mathcal{D}_i$ for the node i.

With these constraints alone, and faced with a source sentence that is particularly difficult to simplify, it is possible for the ILP solver to return a "trivial" solution of no output at all, as all other available solutions result in a negative objective value. It is therefore necessary to impose a global minimum output constraint (1f). In combination with the dependency relations in (1c), this constraint ensures that at least an element of the root sentence is present in the output. Global maximum length constraints are a frequently occurring aspect of ILP models used in NLP applications. We decided not to incorporate any such constraints into our model, as we did not want to place limitations on the simplification of original content.

4 Experimental Setup

In this section we present our experimental setup for assessing the performance of the simplification model described above. We give details on the corpora and grammars we used, model parameters, the systems used for comparison with our approach, and explain how the output was evaluated.

Grammar Extraction QG rules were learned from revision histories and an aligned simplification corpus, which we obtained from snapshots[4] of MainEW and SimpleEW. Wiki-related mark-up and meta-information was removed to extract the plain text from the articles.

SimpleEW revisions not only simplify the text of existing articles, they may also introduce new content, vandalize or remove vandalism, or perform numerous automatic "house-keeping" modifications.

[2] $\text{FKGL} = 0.39 \left(\frac{\text{total words}}{\text{total sentences}}\right) + 1.8 \left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$

[3] Constraints (1b), (1c) and (1d) are shown as dependencies for clarity, but they were implemented as inequalities in the ILP.

[4] The snapshots for MainEW (enwiki) and SimpleEW (simplewiki dated 2010-09-16 and 2010-09-13, respectively (both available from `http://download.wikimedia.org/`).

Corpora	Syntactic	Lexical	Splitting
Revision	316	269	184
Aligned	312	96	254

Table 2: Number of QG rules extracted (after removing singletons) from revision-based and aligned corpora.

We identified suitable revisions for simplification by selecting those where the author had mentioned a keyword (such as *simple*, *clarification* or *grammar*) in the revision comments. Each selected revision was compared to the previous version. Because the entire article is stored at each revision, we needed to identify and align modified sentences. We first identified modified sections using the Unix `diff` program, and then individual sentences within the sections were aligned using the program `dwdiff`[5]. This resulted in 14,831 paired sentences. With regard to the aligned simplification corpus, we paired 15,000 articles from SimpleEW and MainEW following the language link within the snapshot files. Within the paired articles, we identified aligned sentences using macro alignment (at paragraph level) then micro alignment (at sentence level), using tf.idf scores to measure similarity (Barzilay and Elhadad, 2003; Nelken and Schieber, 2006).

All source-target sentences (resulting from revisions or alignments) were parsed with the Stanford parser (Klein and Manning, 2003) in order to label the text with syntactic information. QG rules were created by aligning nodes in these sentences as described earlier. A breakdown of the number and type of rules we obtained from the revision and aligned corpora (after removing rules appearing only once) is given in Table 2. Examples of the most frequently learned QG rules are shown in Table 3. Rules (1)–(3) involve syntactic simplification and rules (4)–(6) involve sentence splitting. Examples of common lexical simplifications found by our grammar are: "*discovered*" → "*found*", "*defeated*" → "*won against*", "*may refer to*" → "*could mean*", "*original*" → "*first*", "*requires*" → "*needs*".

Sentence generation We generated simplified versions of MainEW sentences. For each (parsed) source sentence, we created and solved an ILP (see Equation (1)) parametrized as follows: the number of target words per sentence (wps) was set to 8, and syllables per word (spw) to 1.5. These two parameters were empirically tuned on the training set. To solve the ILP model we used the ZIB Optimization Suite software (Achterberg, 2007; Koch, 2004). The solution was converted into a sentence by removing nodes not chosen from the tree representation, then concatenating the remaining leaf nodes in order.

1.	$\langle \mathrm{S, ST}\rangle \rightarrow \langle [\mathrm{NP}_{\boxed{1}}\ \mathrm{VP}_{\boxed{2}}], [\mathrm{NP}_{\boxed{1}}\ \mathrm{VP}_{\boxed{2}}\ .]\rangle$
2.	$\langle \mathrm{S, ST}\rangle \rightarrow \langle [\mathrm{VP}_{\boxed{1}}], [\textit{This}\ \mathrm{VP}_{\boxed{1}}\ .]\rangle$
3.	$\langle \mathrm{NP, ST}\rangle \rightarrow \langle [\mathrm{NP}_{\boxed{1}}\ , \mathrm{NP}_{\boxed{2}}], [\mathrm{NP}_{\boxed{1}}\ \textit{was}\ \mathrm{VP}_{\boxed{2}}\ .]\rangle$
4.	$\langle \mathrm{ST, ST, ST}\rangle \rightarrow \langle [\mathrm{S}_{\boxed{1}}\ , \textit{and}\ \mathrm{S}_{\boxed{2}}], [\mathrm{ST}_{\boxed{1}}], [\mathrm{ST}_{\boxed{2}}]\rangle$
5.	$\langle \mathrm{ST, ST, ST}\rangle \rightarrow \langle [\mathrm{S}_{\boxed{1}}\ \textit{:}\ \mathrm{S}_{\boxed{2}}], [\mathrm{ST}_{\boxed{1}}], [\mathrm{ST}_{\boxed{2}}]\rangle$
6.	$\langle \mathrm{ST, ST, ST}\rangle \rightarrow \langle [\mathrm{S}_{\boxed{1}}\ , \textit{but}\ \mathrm{S}_{\boxed{2}}], [\mathrm{ST}_{\boxed{1}}], [\mathrm{ST}_{\boxed{2}}]\rangle$

Table 3: Examples of QG rules involving syntactic simplification (1)–(3) and sentence division (4)–(6). The latter are shown as the tuple ⟨source, target, aux⟩. The transform of nodes from S to ST (for example) rely on the application of syntactic simplification rules rules. Boxed subscripts show aligned nodes.

Evaluation We evaluated our model on the same dataset used in Zhu et al. (2010), an aligned corpus of MainEW and SimpleEW sentences. The corpus contains 100/131 source/target sentences and was created automatically. Sentences from this corpus (and their revisions) were excluded from training. We evaluated two versions of our model, one with rewrite rules acquired from revision histories of simplified documents and another one with rules extracted from MainEW-SimpleEW aligned sentences. These models were compared against Zhu et al. (2010)[6] who also learn simplification rules from Wikipedia, and a simple baseline that uses solely lexical simplifications[7] provided by the SimpleEW editor "SpencerK" (Spencer Kelly). An obvious idea would be to treat sentence simplification as an English-to-English translation problem and use an off-the-shelf system like Moses[8] for the task. However, we refrained from doing so as Zhu et al. (2010) show that Moses performs poorly, it cannot model rewrite operations that split sentences or drop words and in most cases generates output identical

[5] `http://os.ghalkes.nl/dwdiff.html`

[6] We are grateful to Zhemin Zhu for providing us with his test set and the output of his system.

[7] `http://www.spencerwaterbed.com/soft/simple/`

[8] `http://www.statmt.org/moses/`

MainEW	Wonder has recorded several critically acclaimed albums and hit singles, and writes and produces songs for many of his label mates and outside artists as well.
Zhu et al	Wonder has recorded several praised albums and writes and produces songs. Many of his label mates and outside artists as well.
AlignILP	Wonder has recorded several critically acclaimed albums and hit singles. He produces songs for many of his label mates and outside artists as well. He writes.
RevILP	Wonder has recorded many critically acclaimed albums and hit singles. He writes. He makes songs for many of his label mates and outside artists as well.
SimpleEW	He has recorded 23 albums and many hit singles, and written and produced songs for many of his label mates and other artists as well.
MainEW	The London journeys In 1790, Prince Nikolaus died and was succeeded by a thoroughly unmusical prince who dismissed the entire musical establishment and put Haydn on a pension.
Zhu et al	The London journeys in 1790, prince Nikolaus died and was succeeds by a son became prince. A son became prince told the entire musical start and put he on a pension.
AlignILP	The London journeys In 1790, Prince Nikolaus died. He was succeeded by a thoroughly unmusical prince. He dismissed the entire musical establishment. He put Haydn on a pension.
RevILP	The London journeys In 1790, Prince Nikolaus died. He was succeeded by a thoroughly unmusical prince. He dismissed the whole musical establishment. He put Haydn on a pension.
SimpleEW	The London journeys In 1790, Prince Nikolaus died and his son became prince. Haydn was put on a pension.

Table 4: Example simplifications produced by the systems in this paper (RevILP, AlignILP) and Zhu et al.'s (2010) model, compared to real Wikipedia text (MainEW: input source, SimpleEW: simplified target).

to the source.

We evaluated model output in two ways, using automatic evaluation measures and human judgments. Intuitively, readability measures ought to be suitable for assessing the output of simplification systems. We report results with the well-known Flesch-Kincaid Grade Level index (FKGL). Experiments with other readability measures such as the Flesch Reading Ease and the Coleman-Liau index obtained similar results. In addition, we also assessed how the system output differed from the human SimpleEW gold standard by computing BLEU (Papineni et al., 2002) and TERp (Snover et al., 2009). Both measures are commonly used to automatically evaluate the quality of machine translation output. BLEU[9] scores the target output by counting *n*-gram matches with the reference, whereas TERp is similar to word error rate, the only difference being that it allows shifts and thus can account for word order differences. TERp also allows for stem, synonym, and paraphrase substitutions which are common rewrite operations in simplification.

In line with previous work on text rewriting (e.g., Knight and Marcu 2002) we also evaluated system output by eliciting human judgments. We conducted three experiments. In the first experiment participants were presented with a source sentence and its target simplification and asked to rate whether the latter was easier to read compared to the source. In the second experiment, they were asked to rate the grammaticality of the simplified output. In the third experiment, they judged how well the simplification preserved the meaning of the source. In all experiments participants used a five point rating scale where a high number indicates better performance. We randomly selected and automatically simplified 64 sentences from Zhu et al.'s (2010) test corpus using the four models described above. We also included gold standard simplifications. Our materials thus consisted of 320 (64 × 5) source-target sentences.[10] We collected ratings from 45 unpaid volunteers, all self reported native English speakers. The studies were conducted over the Internet using a custom built web interface. Examples of our experimental items are given in Table 4 (we omit the output of SpencerK as this is broadly similar to the source sentence, modulo lexical substitutions).

[9] We calculated single-reference BLEU using the *mteval-v13a* script (with the default settings).

[10] A Latin square design ensured that subjects did not see two different simplifications of the same sentence.

Models	FKGL	BLEU	TERP
MainEW	15.12	—	—
SimpleEW	11.25	—	—
SpencerK	14.67	0.47	0.51
Zhu et al	9.41	0.38	0.59
RevILP	10.92	0.42	0.60
AlignILP	12.36	0.34	0.85

Table 5: Model performance using automatic evaluation measures.

5 Results

The results of our automatic evaluation are summarized in Table 5. The first column reports the FKGL readability index of the source sentences (MainEW), of their target simplifications (SimpleEW) and the output of four models: a simple baseline that relies on lexical substitution (SpencerK), Zhu et al.'s (2010) model, and two versions of our model, one trained on revision histories (RevILP) and another one trained on the MainEW-SimpleEW aligned corpus (AlignILP). As can be seen, the source sentences have the highest reading level. Zhu et al.'s system has the lowest reading level followed by our own models and SpencerK. All models are significantly[11] different in reading level from SimpleEW with the exception of RevILP (using a one-way ANOVA with post-hoc Tukey HSD tests). SpencerK is not significantly different in readability from MainEW; RevILP is significantly different from Zhu et al. and AlignILP. In sum, these results indicate that RevILP is the closest to SimpleEW and that the provenance of the QG rules has an impact on the model's performance.

Table 5 also shows BLEU and TERp scores with SimpleEW as the reference. These scores can be used to examine how close to the gold standard our models are. SpencerK has the highest BLEU and lowest TERp scores.[12] This is expected as this baseline performs only a very limited type of rewriting, namely lexical substitution. AlignILP is most different from the reference, followed by Zhu et al. (2010) and RevILP. Taken together these results indicate that the ILP models perform a fair amount of rewriting without simply rehashing the source sentence.

Models	Simplicity	Grammaticality	Meaning
SimpleEW	3.74	4.89	4.41
SpencerK	1.41	4.87	4.84
Zhu et al	2.92	3.43	3.44
RevILP	3.64	4.55	4.19
AlignILP	2.69	4.03	3.98

Table 6: Average human ratings for gold standard SimpleEW sentences, a simple baseline (SpencerK) based on lexical substitution, Zhu et al.'s 2010 model, and two versions of our ILP model (RevILP and AlignILP).

	Zhu et al	AlignILP	RevILP	SimpleEW
SpencerK	□◇△	□◇△	□◆△	□◆△
Zhu et al		■◇△	□◇△	□◇△
AlignILP			□◇▲	□◇△
RevILP				■◆▲

Table 7: □/■: is/not sig. diff. wrt simplicity; ◇/◆: is/not sig. diff. wrt grammaticality; △/▲: is/not sig. diff. wrt meaning.

We now turn to the results of our judgment elicitation study. Table 6 reports the average ratings for Simplicity (is the target sentence simpler than the source?), Grammaticality (is the target sentence grammatical?), and Meaning (does the target preserve the meaning of the source?). With regard to simplicity, our participants perceive the gold standard (SimpleEW) to be the simplest, followed by RevILP, Zhu et al, and AlignILP. SpencerK is the least simple model and the most grammatical one as lexical substitutions do not change the structure of the sentence. Interestingly, RevILP and AlignILP are also rated highly with regard to grammaticality. Zhu et al. (2010) is the least grammatical model. Finally, RevILP preserves the meaning of the target as well as SimpleEW, whereas Zhu et al. yields the most distortions. Again SpencerK is rated highly amongst the other models as it is does not substantially simplify and thus change the meaning of the source.

Table 7 reports on pairwise comparisons between all models and their statistical significance (again using a one-way ANOVA with post-hoc Tukey HSD tests). RevILP is not significantly different from SimpleEW on any dimension (Simplicity, Grammat-

[11] All significance differences reported throughout this paper are with a level less than 0.01.

[12] The perfect BLEU score is one and the perfect TERp score is zero.

Original story: There was once a sweet little maid *who* lived with her father and mother in a pretty little cottage at the edge of the village. At the further end of the wood *was another pretty cottage and in it* lived her grandmother. Everybody loved this little *girl,* her grandmother perhaps loved her most of all *and gave* her a great many pretty things. Once she gave her a red cloak with a hood *which she always wore,* so *people* called her Little Red Riding Hood.
Generated simplification: There was once a sweet little maid. She lived with her father and mother in a pretty little cottage at the edge of the village. At the further end of the wood it lived her grandmother. Everybody loved this little girl. Her grandmother perhaps loved her most of all. She gave her a great many pretty things. Once she gave her a red cloak with a hood, so persons called her Little Red Riding Hood.

Table 8: Excerpt of *Little Red Riding Hood* simplified by the RevILP model. Modifications to the original story are highlighted in italics.

icality, Meaning), whereas Zhu et al. differs significantly from RevILP and SimpleEW on all dimensions. It is also significantly different from AlignILP in terms of grammaticality and meaning but not simplicity. RevILP is significantly more simple and grammatical than AlignILP but performs comparably with respect to preserving the meaning of the source.

In sum, our results show that RevILP is the best performing model. It creates sentences that are simple, grammatical and adhere to the meaning of the source. The QG rules obtained from the revision histories produce better output compared to the aligned corpus. As revision histories are created by Wikipedia contributors, they tend to be a more accurate data source than aligned sentences which are obtained via an automatic and unavoidably noisy procedure. Our results also show that a more general model not restricted to specific rewrite operations like Zhu et al. (2010) obtains superior results and has better coverage.

We also wanted to see whether a simplification model trained on Wikipedia could be applied to another domain. To this end, we used RevILP to simplify five children stories from the Gutenburg[13] collection. The model simplified one sentence at a time and was ran with the Wikipedia settings without any modification. The mean FKGL on the simplified stories was 3.78. compared to 7.04 for the original ones. An example of our system's output on *Little Red Riding Hood* is shown in Table 8.

Possible extensions and improvements to the current model are many and varied. We have presented an all-purpose simplification model without a target audience or application in mind. An interesting research direction would be to simplify text according to readability levels or text genres (e.g., newspaper vs literary text). We could do this by incorporating readability-specific constraints to the ILP or by changing the objective function (e.g., by favoring more domain-specific rules). Finally, we would like to extend the current model so as to simplify entire documents both in terms of style and content.

Acknowledgments We are grateful to Lillian Lee whose invited talk at CoNLL-2010 inspired this research. We would also like to thank the members of the Probabilistic Models of Language group at the School of Informatics for valuable discussions and comments. We acknowledge the support of EPSRC through project grant EP/F055765/1.

References

Achterberg, Tobias. 2007. *Constraint Integer Programming*. Ph.D. thesis, Technische Universität Berlin.

Barzilay, Regina. 2003. *Information Fusion for Multi-Document Summarization: Paraphrasing and Generation*. Ph.D. thesis, Columbia University.

Barzilay, Regina and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan, pages 25–32.

Beigman Klebanov, Beata, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *Proceedings of Ontologies, Dabases, and Applications of Semantics (ODBASE) International Conference*.

[13] http://www.gutenberg.org

Springer, Agia Napa, Cyprus, volume 3290 of *Lecture Notes in Computer Science*, pages 735–747.

Carroll, John, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the ACL*. Bergen, Norway, pages 269–270.

Chandrasekar, Raman, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics*. Copenhagen, Denmark, pages 1041–1044.

Cohn, Trevor and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, UK, pages 137–144.

Das, Dipanjan and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the ACL-IJCNLP*. Suntec, Singapore, pages 468–476.

Devlin, Siobhan. 1999. *Simplifying Natural Language for Aphasic Readers*. Ph.D. thesis, University of Sunderland.

Inui, Kentaro, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*. Association for Computational Linguistics, Sapporo, Japan, pages 9–16.

Kaji, Nobuhiro, Daisuke Kawahara, Sadao Kurohashi, and Satoshi Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 215–222.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*. Sapporo, Japan, pages 423–430.

Knight, Kevin and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.

Koch, Thorsten. 2004. *Rapid Mathematical Prototyping*. Ph.D. thesis, Technische Universität Berlin.

Mitchell, James V. 1985. *The Ninth Mental Measurements Year-book*. University of Nebraska Press, Lincoln, Nebraska.

Nastase, Vivi and Michael Strube. 2008. Decoding Wikipedia categories for knowledge acquisition. In *Proceedings of the 23rd Conference on Artificial Intelligence*. pages 1219–1224.

Nelken, Rani and Stuart Schieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy, pages 161–168.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*. Philadelphia, PA, pages 311–318.

Ponzetto, Simone Paolo and Michael Strube. 2007. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research* 30:181–212.

Sauper, Christina and Regina Barzilay. 2009. Automatically generating Wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, Suntec, Singapore, pages 208–216.

Siddharthan, Advaith. 2003. *Syntactic Simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge, University of Cambridge.

Siddharthan, Advaith. 2004. Syntactic simplification and text cohesion. in research on language and computation. *Research on Language and Computation* 4(1):77–109.

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

tion. Association for Computational Linguistics, New York City, pages 23–30.

Smith, David A. and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of the EMNLP*. Suntec, Singapore, pages 822–831.

Snover, Matthew, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece, pages 259–268.

Vickrey, David and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, pages 344–352.

Wang, Mengqiu, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the EMNLP-CoNLL*. Prague, Czech Republic, pages 22–32.

Watanabe, Willian Massami, Arnaldo Candido Junior, Vinícius Rodriguez de Uzẽda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Alu´sio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of Communication*. Bloomington, IN.

Woodsend, Kristian, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Cambridge, MA, pages 513–523.

Wu, Fei and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, pages 118–127.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*. Toulouse, France, pages 523–530.

Yamangil, Elif and Rani Nelken. 2008. Mining Wikipedia revision histories for improving sentence compression. In *Proceedings of ACL-08: HLT, Short Papers*. Association for Computational Linguistics, Columbus, Ohio, pages 137–140.

Yatskar, Mark, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*. pages 365–368.

Zhao, Shiqi, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Singapore, pages 834–842.

Zhu, Zhemin, 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*. Beijing, China, pages 1353–1361.